UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

THIAGO CARVALHO SOUSA

Simulação computacional do sangue usando o método *Smoothed Particle
Hydrodynamics* (SPH)

São Paulo

2020

THIAGO CARVALHO SOUSA

Simulação computacional do sangue usando o método *Smoothed Particle Hydrodynamics* (SPH)

Versão corrigida

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo como parte dos requisitos para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 29 de novembro de 2019. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Helton Hideraldo Bísaro

São Paulo

2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)
CRB 8 - 4936

Sousa, Thiago Carvalho

Simulação computacional do sangue usando o método
Smoothed Particle Hydrodynamics (SPH) / Thiago Carvalho Sousa ;
orientador, Helton Hideraldo Bísaro. – 2020.

93 f. : il

Dissertação (Mestrado em Ciências) - Programa de Pós-
Graduação em Sistemas de Informação, Escola de Artes,
Ciências e Humanidades, Universidade de São Paulo, em
2019.

Versão corrigida

1. Realidade virtual. 2. Sangue - Simulação
computacional. 3. Métodos numéricos. 4. *Smoothed Particle*
Hydrodynamics. I. Bísaro, Helton Hideraldo, orient. II. Título

CDD 22.ed.– 006.8

Dissertação de autoria de Thiago Carvalho Sousa, sob o título “**Simulação computacional do sangue usando o método *Smoothed Particle Hydrodynamics* (SPH)**”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 29 de novembro de 2019 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Helton Hideraldo Bísaro

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo
Presidente

Prof. Dr. Cássio Machiaveli Oishi

Faculdade de Ciências e Tecnologia - Universidade Estadual Paulista

---

Prof. Dr. Marcelo de Souza Lauretto

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo

---

Profa. Dra. Karla Roberta Pereira Sampaio Lima

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo

*Dedico este trabalho à memória de meus queridos tios
Ana Maria Sousa, Carmem Conceição Sousa, Maiana Moura e José Donizete Sousa;
que faleceram enquanto este trabalho estava sendo produzido.*

Dedico, também, à todos os alunos que se graduaram por meio de uma Licenciatura.

Agradecimentos

Agradeço primeiramente a Deus, pelo dom da vida; e a intercessão de Nossa Senhora de Lourdes, que me possibilitou chegar até aqui.

Agradeço imensamente ao meu orientador, Prof. Helton, pela paciência e pelos valiosos ensinamentos ao longo dessa jornada.

Aos meus pais Rita e Sinval, meus irmãos Fernanda e André, e meus pequenos sobrinhos Aurora e Plínio; agradeço pelo incentivo, apoio e compreensão de sempre!

Aos tios Tereza e Bira, primas Cibele e Celina, meu afilhado Pedro Henrique, e todos os demais familiares; eu agradeço pelo carinho e suporte!

Agradeço às Profas. Gisele e Claudia, da EACH-USP, pelas primeiras orientações de Mestrado. E as Profas. Iole e Mariane Streibel, do IME-USP, agradeço pela iniciação nesse fabuloso mundo da pesquisa científica.

Aos Profs. Marcelo e Fátima, agradeço os relevantes *feedbacks* e sugestões dadas no Exame de Qualificação. E ao Prof. André Gal, agradeço pela supervisão e as dicas durante o estágio PAE de docência no ensino superior.

Aos companheiros do Mestrado: Daniel, Tharsis, Eliana, Débora, Wanderson, Bruno, Micael, Gabriel, Antônio, Edson, Keith e Rafael, agradeço pela troca de experiências. E aos colegas do Laboratório de Aplicações de Informática em Saúde (LApIS), agradeço pelas parcerias e networking.

Aos amigos do IME, Thiago, Noele e Lilian; agradeço pela amizade duradoura. E aos meus amigos Tatiane, Marco, Amanda, Cauan, Juliene, Luiz Tadeu, Mário e Adriano, do Jardim Clipper; agradeço pela amizade e o companheirismo de longa data!

Gostaria de agradecer também à Roberta, Renata, Eliane, Nataniel, Adauto, Carla, Fernando César, Raffael, Fabíola, Caio, Daniele, Aline, Jefferson, Leonardo, Equipe Infor e demais colegas do Banco do Brasil, que me auxiliaram (dentro do possível) a conciliar as atividades do Mestrado com as do trabalho.

Em suma, quero agradecer à todas as pessoas e instituições que de alguma forma contribuíram para a realização deste trabalho! Meu muito obrigado!!!

“Nunca deixe que lhe digam que não vale a pena acreditar no sonho que se tem...”

(Renato Russo e Flávio Venturini)

Resumo

SOUSA, Thiago Carvalho. **Simulação computacional do sangue usando o método *Smoothed Particle Hydrodynamics* (SPH)** 2020. 93 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2019.

Simular computacionalmente fluidos biológicos é relevante para o treinamento médico, sistemas de apoio à decisão, desenvolvimento de novos equipamentos médicos, investigação de doenças, entre outras aplicações. Apesar disso, nossa revisão sistemática apontou que há carência de sistemas nesse sentido; e os existentes apresentam lacunas como o uso em regiões de geometrias complexas, e elevado tempo de execução. O objetivo de nosso trabalho é usar e aperfeiçoar o método numérico SPH (*Smoothed Particle Hydrodynamics*) para adaptar uma ferramenta computacional para simulação do sangue, com o objetivo de aprimorar o realismo de tais simulações e aperfeiçoar o teste de colisão com fronteiras - em nosso caso, se tratam de vasos sanguíneos representados por uma malha geométrica composta de triângulos. Neste trabalho apresentamos os fundamentos do método SPH; as características de sua implementação computacional; os resultados da revisão sistemática; as adaptações feitas para manter a incompressibilidade do fluido; os testes de validação numérica baseados no fluxo de Poiseuille; e a proposição de um novo algoritmo de colisão, que tem por base rotações 3D e números quatérnios. Nossos resultados mostraram que o novo teste de colisão foi efetivo, eliminando totalmente casos onde o fluido “escapava” do vaso sanguíneo.

Palavras-chaves: SPH (Smoothed Particle Hydrodynamics). Sangue. Métodos numéricos. Simulação computacional. Fluxo de Poiseuille. Fronteiras. Malha triangular.

Abstract

SOUSA, Thiago Carvalho. **Computational simulation of blood using the Smoothed Particle Hydrodynamics (SPH) method** 2020. 93 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2019.

Computational simulation of biological fluids is relevant for medical training, decision support systems, development of new medical equipment, disease research, among other applications. Nevertheless, our systematic review has indicated that there is a lack of systems in this regard; and existing ones gaps such as the use in regions of complex geometries, and high execution time. The objective of our work is to use and improve the numerical method SPH (Smoothed Particle Hydrodynamics) to adapt a computational tool for blood simulation, in order to improve the realism of such simulations and meliorate the boundary collision test - in our case, they are blood vessels represented by a geometric mesh composed of triangles. In this paper we present the fundamentals of the SPH method; the characteristics of its computational implementation; the results of the systematic review; adaptations made to maintain fluid incompressibility; numerical validation tests based on Poiseuille flow; and the proposition of a new collision algorithm, based on 3D rotations and quaternion numbers. Our results showed that the new boundary collision test was effective, completely eliminating cases where fluid “escaped” from the blood vessel.

Keywords: SPH (Smoothed Particle Hydrodynamics). Blood flow. Numerical methods. Simulation computational. Poiseuille flow. Boundaries. Triangular mesh.

Lista de figuras

Figura 1 – Finalidades dos artigos	29
Figura 2 – Significado da Função Núcleo. Na figura (a) vemos as aproximações por partículas, usando partículas dentro do raio suporte da função de suavização W para a partícula i . A busca por vizinhança possui formato circular com raio kh . Na figura (b), as alturas das colunas ilustram o “peso” que cada partícula exerce na partícula de interesse (inversamente proporcional à distância).	39
Figura 3 – Ilustração do esquema de integração numérica Leap-frog.	50
Figura 4 – Simulador SPH em execução com o fluido água, em 3 momentos: na figura (a), observamos o fluido discretizado em partículas. No momento (b), aplicamos uma das técnicas de visualização de superfícies (seção 4.11); e finalmente em (c) observamos a água que “bate” na taça vermelha (modelada por uma malha triangular) e é “derramada”, testando o algoritmo de colisão com fronteiras (seção 4.9)	55
Figura 5 – Princípio das partículas fantasmas	56
Figura 6 – Fronteira rígida com partículas fantasmas	57
Figura 7 – Colisão de uma esfera (partícula) com um triângulo (fronteira)	58
Figura 8 – Teste de colisão; a) tempo de colisão t_c ; b) colisão com um dos vértices de T ; c) colisão com o interior de T ; d) colisão com uma das arestas de T	59
Figura 9 – Ponto r , que determina se a esfera e_i intersecta a aresta V_tW_t	60
Figura 10 – Comparação de diferentes técnicas de visualização para o mesmo instante de uma simulação. Em (a), temos o fluido discretizado em partículas. Em (b), a superfície foi gerada considerando a técnica de Müller, Charypar e Gross (2003), em (c) por Zhu e Bridson (2005), em (d) por Onderik, Chládek e Durikovic (2013), em (e) por Yu e Turk (2013) e em (f) por Sandim et al. (2016).	65
Figura 11 – Simulador SPH em funcionamento com o fluido sangue, em 4 momentos: na figura (a) temos o início da simulação, que passa pelos instantes intermediários (b) e (c) e finaliza em (d). As imagens finais foram geradas com um algoritmo de traçado de raios.	66

Figura 12 – Arquivo .OFF do vaso sanguíneo com uma bifurcação, modelado por malha triangular. A visualização foi obtida pelo software MeshLab. . .	67
Figura 13 – Representação gráfica do Teste de Fluxo de Poiseuille	69
Figura 14 – Exemplo de fluido laminar	70
Figura 15 – Exemplo de fluxo turbulento	70
Figura 16 – Comparação entre os resultados observados e os calculados, considerando $V = 0,01m/s$	72
Figura 17 – Comparação entre os resultados observados e os calculados, considerando $V = 0,1m/s$	73
Figura 18 – A partícula i em rota de colisão com a aresta de um triângulo da malha (a). A velocidade e a posição da partícula é calculada (b). A partícula tem o seu curso alterado, mas a nova posição escapa do triângulo de baixo (c). A partícula i “vazou” da malha.	74
Figura 19 – Simulação Dam Break. Esta simulação tinha ao todo 15.348 partículas, mas 39 delas “vazaram” da malha pelas arestas dos triângulos.	75
Figura 20 – Uma partícula i do fluido, com velocidade v_i , em rota de colisão com o triângulo T_j da malha.	76
Figura 21 – Transportar os vetores para a Origem.	77
Figura 22 – Aplicar a sequência de rotações ao par (triângulo T_j + partícula i). . .	77
Figura 23 – Teste em uma caixa cúbica, usando o Algoritmo Tradicional (a) e o Algoritmo de Colisão Rotacionada (b). As cores das partículas indicam a intensidade de suas velocidades.	80
Figura 24 – Teste no vaso sanguíneo com 1 bifurcação. Em (a), foi usado o Algoritmo Tradicional. Note a quantidade de partículas que “escapam”, especialmente na bifurcação do meio. Em (b) foi usado o Algoritmo de Colisão Rotacionada. Observe que não são encontrados “vazamentos”. .	81
Figura 25 – Teste em um vaso com 3 bifurcações. Em (a), usando o Algoritmo Tradicional, pode ser visto que algumas partículas acabam “escapando” da malha triangular. Em (b), foi usado o Algoritmo de Colisão Rotacionada. Em ambos os casos, as cores das partículas indicam a intensidade de suas velocidades.	82

Lista de algoritmos

Algoritmo 1 – Algoritmo básico do método SPH	53
Algoritmo 2 – Algoritmo básico de colisão	60
Algoritmo 3 – Teste de Colisão Rotacionada	78

Lista de tabelas

Tabela 1 – Parâmetros usados em nossa implementação	64
Tabela 2 – Resultados do Teste para vazamento de partículas, comparando o algoritmo originalmente utilizado (baseado na obra de Neto (2007)), e o Algoritmo de Colisão Rotacionada que implementamos.	83

Lista de abreviaturas e siglas

2D	2 Dimensões
3D	3 Dimensões
API	Application Programming Interface
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CGAL	Computational Geometry Algorithms Library
CPU	Central Process Unit
CUDA	Code Unified Device Architecture
DEM	Diffuse Element Method
EDP	Equação de Derivadas Parciais
EDO	Equação Diferencial Ordinária
EFGM	Element Free Galerkin Method
fps	frames per second
GPU	Graphics Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
ISPH	truly Incompressible SPH
MLPG	Meshless Local Petrov-Galerkin
OFF	Object File Format
OpenGL	Open Graphics Library
RA	Realidade Aumentada
RS	Revisão Sistemática
RV	Realidade Virtual

SPH	Smoothed Particle Hydrodynamics
WCSPH	Weakly or quasi-Compressibility SPH
XSPH	eXtend Smoothed Particles Hydrodynamics

Lista de símbolos

$\frac{\partial}{\partial t}$	derivada parcial
$\nabla \cdot \mathbf{v}$	divergente do campo vetorial $\mathbf{v}$
t	tempo
$\mathbf{v}$	vetor velocidade
ρ	densidade
p	pressão
$\mathbf{g}$	vetor de aceleração da gravidade
$\mathbf{S}$	tensor extra-tensão
∇	operador de gradiente
μ	coeficiente de viscosidade do fluido
$\mathbf{D}$	tensor taxa de deformação
$\mathbf{I}$	tensor identidade
$\mathbb{R}$	conjunto dos Números Reais
Ω	domínio
δ	função Delta de Dirac
W	função kernel de suavização, função núcleo
h	comprimento do raio de suavização
$\langle \rangle$	aproximação
Co	número de Courant
$ U $	norma da magnitude de velocidade local
Δt	intervalo de tempo
$\frac{d(\ )}{dt}$	derivada material

ρ_0	densidade de referência do fluido.
c	velocidade do som
$\mathbf{v}_i$	vetor velocidade da partícula i
$\mathbf{x}_i$	vetor posição da partícula i
∇^2	operador Laplaciano
π	constante matemática, aproximadamente 3,14159265359
Γ_{ig}	força de repulsão
t_c	tempo de colisão
Ma	número de Mach
Q	vazão do tubo cilíndrico
R	raio
l	comprimento do tubo
η	coeficiente de viscosidade
L	diâmetro do tubo
Re	número de Reynolds
V	velocidade máxima estável do fluido
L	diâmetro do tubo
ν	viscosidade cinemática do fluido
$v_{i,SPH}$	velocidade observada pelo simulador SPH para a partícula i
$v_{i,Exato}$	velocidade calculadas pela solução analítica para a partícula i
$\mathbb{H}$	conjunto dos quatérnios
$\mathbb{C}$	conjunto dos números complexos
q	representação de um número quatérnio

$\bar{q}$	conjugado de um quatérnio
$ q $	magnitude de um quatérnio
$R_q(p)$	operador de rotação com quatérnio
θ	ângulo de rotação
i, j, k	unidades imaginárias dos quatérnios
$\vec{v}$	vetor v

Sumário

1	Introdução	21
1.1	<i>Motivação . . . . .</i>	22
1.2	<i>Tema da Pesquisa . . . . .</i>	23
1.3	<i>Problema e Lacuna de Pesquisa . . . . .</i>	23
1.4	<i>Objetivo . . . . .</i>	23
1.5	<i>Hipóteses e Justificativa . . . . .</i>	24
1.6	<i>Metodologia . . . . .</i>	24
1.7	<i>Organização do trabalho . . . . .</i>	25
2	Revisão Bibliográfica	26
2.1	<i>Metodologia da Revisão . . . . .</i>	26
2.1.1	Protocolo da Revisão Sistemática	27
2.2	<i>Resultados . . . . .</i>	28
2.2.1	Variedade de aplicações	28
2.2.2	Colisões em Fronteiras	28
2.2.3	Diferentes maneiras de fazer a Busca por Partículas Vizinhas e Tratar colisões	30
2.2.4	Aceleração	31
2.2.5	Simulações em 2D x 3D	31
2.2.6	Simulação de Várias Fases	32
2.2.7	Interação com dispositivos não convencionais	33
2.2.8	Aquisição de dados, validação e valores específicos	33
2.3	<i>Conclusões . . . . .</i>	34
3	Conceitos Fundamentais	36
3.1	<i>Fundamentos do Método . . . . .</i>	38
3.2	<i>Aproximação por partículas . . . . .</i>	42
3.3	<i>Condição CFL . . . . .</i>	43
3.4	<i>Números Quatérnios . . . . .</i>	44
3.4.1	Rotações em 3D com Quatérnios	45

4	Aspectos Computacionais	47
4.1	<i>Formulação Lagrangeana das Equações de Navier-Stokes . . . . .</i>	47
4.2	<i>Cálculo da pressão . . . . .</i>	48
4.3	<i>Integração temporal . . . . .</i>	48
4.4	<i>Funções núcleo . . . . .</i>	50
4.5	<i>Estabilidade numérica . . . . .</i>	52
4.6	<i>Suavização da Velocidade XSPH . . . . .</i>	52
4.7	<i>Algoritmo geral . . . . .</i>	53
4.8	<i>Simulador SPH . . . . .</i>	54
4.9	<i>Condição de fronteira e Teste de colisão . . . . .</i>	54
4.10	<i>Incompressibilidade . . . . .</i>	61
4.10.1	WCSPH	62
4.11	<i>Visualização de superfícies . . . . .</i>	63
4.12	<i>Parâmetros da Simulação . . . . .</i>	64
4.13	<i>Simulação de uso . . . . .</i>	64
5	Validação do simulador	68
5.1	<i>Número de Reynolds . . . . .</i>	69
5.2	<i>Descrição do Teste . . . . .</i>	70
5.3	<i>Resultados e discussões . . . . .</i>	71
6	Teste de Colisão com Rotações	74
6.1	<i>Descrição do novo teste de colisão . . . . .</i>	76
6.2	<i>Teste de Colisão Rotacionada . . . . .</i>	78
6.3	<i>Resultados . . . . .</i>	79
6.4	<i>Discussões Finais . . . . .</i>	83
7	Conclusões	85
7.1	<i>Trabalhos Futuros . . . . .</i>	86
7.1.1	Paralelização do simulador	86
7.1.2	Visualização de superfícies	87
7.1.3	Malhas elásticas	87
7.1.4	Simulação multi-fase	87
7.1.5	Interação com dispositivos hápticos	88

Referências ¹	89

---

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Após a segunda metade do século XX, ocorreu um notável avanço dos computadores e da sua disponibilidade, o que estimulou o desenvolvimento e a implementação de códigos computacionais direcionados, por exemplo, às simulações numéricas de fluidos (FRIGO, 2004).

Tradicionalmente, os métodos numéricos para simulação de fenômenos físicos, baseiam-se na discretização do domínio de cálculo em malhas. Entretanto, frente à limitação de representação de domínios de geometrias complexas, descontinuidades ou superfícies móveis, aplicam-se algumas técnicas para contornar tais inconvenientes, como a reestruturação de malha ou malhas adaptativas. Apesar de conseguir tratar as descontinuidades do domínio, a contrapartida é que se aumenta consideravelmente o tempo de processamento, além de atrelar a confiabilidade da solução obtida ao algoritmo de reestruturação de malha (PINTO, 2013).

Novos métodos numéricos, desvinculados do paradigma de discretização da malha, vêm ganhando força nos últimos anos, principalmente pela capacidade em representar fenômenos altamente não-lineares (como o sangue) e a naturalidade com a qual tratam descontinuidades. Estes métodos são conhecidos como métodos de partículas - de uma maneira geral, métodos numéricos de solução de um problema físico que não necessitam de malhas no domínio a ser simulado (*meshfree*).

Este domínio é representado por um conjunto de partículas, cuja dinâmica é determinada de acordo com suas equações de movimento escritas na forma Lagrangiana. Dentre os métodos de partículas destaca-se o SPH (*Smoothed Particle Hydrodynamics*) (VASCO, 2014).

No método numérico SPH não há necessidade de serem geradas malhas: a discretização é realizada sob uma quantidade finita de partículas, e o método é desenvolvido sob uma teoria de interpolação de dados, - sendo esse, aliás, um dos seus fundamentos. Para calcular os valores das grandezas físicas de interesse são utilizadas as “funções de suavização”, interpolando os valores existentes dentro de uma vizinhança fechada, que é determinada por um raio de suporte compacto - que se localiza em torno de um ponto material de interesse (uma partícula do fluido). O estado de cada partícula dependerá do estado em que se encontram as “partículas vizinhas”.

1.1 Motivação

O campo da realidade virtual - particularmente, na simulação computacional do sangue - tem apresentado importantes contribuições para a área médica. Ambientes virtuais de treinamento médico, sistemas de apoio a decisão para especialistas, e o desenvolvimento de novos equipamentos médicos são algumas das aplicações já desenvolvidas (RIBEIRO; NUNES, 2014; BÍSCARO et al., 2016; TORI et al., 2011; XIAO et al., 2017; AL-SAAD; KULASEGARAM; BORDAS, 2016).

Como exemplo, podemos citar uma sub-área da Realidade Virtual conhecida como “Cirurgia Virtual” (SHI; ZHENG; LIU, 2017). Essa área busca fornecer ambientes de treinamento imersivos para cirurgiões. É intuitivo o fato de que, num procedimento cirúrgico, sangramentos são naturais e até mesmo inevitáveis, devido aos cortes nos tecidos. Dessa forma, simulações de sangramento podem não apenas aumentar o realismo dos ambientes de treinamento, mas também podem aumentar a habilidade do cirurgião para lidar com este fato.

Nesse mesmo contexto, o trabalho de Holland, Pop e John (2017) trabalha com a hipótese de que treinamento e suporte durante uma intervenção cirúrgica cardíaca podem ser significativamente melhorados com o uso de ferramentas que envolvem Realidade Virtual, Realidade Mista e simulações computacionais. De acordo com os autores, os cirurgiões devem ser capazes de interagir com a geometria da região em análise, de maneira a investigar como o fluxo será afetado pelo tratamento. Isso implicaria num sistema capaz de rodar diferentes cenários num tempo útil aos profissionais, o que proporcionaria informações valiosas para o planejamento da cirurgia.

Outro exemplo de aplicação possível é a análise de anomalias no sistema circulatório humano, tal como a trombose (TANAKA; TAKANO, 2005), que pode ser a causa de diversas doenças no corpo humano. Com o aumento da capacidade computacional, sistemas - tanto de treinamento virtual, quanto de suporte à decisões/diagnóstico - têm sido cada vez mais utilizados e estão sendo difundidos na comunidade médica.

Portanto, simular computacionalmente o sangue é extremamente relevante e contribui no aprimoramento da medicina, seja no treinamento ou preparação/desenvolvimento de novos tratamentos.

1.2 Tema da Pesquisa

Localizado na área de Computação Gráfica, o tema geral de nosso trabalho é o uso do método SPH (*Smoothed Particle Hydrodynamics*) para realizar uma simulação computacional aplicada à saúde (do fluido biológico sangue), bem como propor contribuições ao método.

1.3 Problema e Lacuna de Pesquisa

Apesar da simulação computacional ser amplamente aplicada a diversas áreas, profissionais da saúde relatam carência de sistemas nesse sentido. Em especial, reclamam que sistemas que simulam o sangue humano são de “baixa qualidade”, com elevado tempo de execução (YE; PHAN-THIEN; LIM, 2016).

Mesmo existindo outros métodos numéricos para simulação computacional de fluidos como o sangue (como apontado na revisão de Ye, Phan-Thien e Lim (2016)), o método SPH tem a vantagem de não depender de malhas. Ou seja, economiza recursos computacionais que seriam usados na geração, armazenamento e manutenção de malhas geométricas, o que resultará diretamente numa melhora de performance.

Mesmo com esta importante vantagem, em contrapartida, o uso desta técnica exige que se calcule a vizinhança para cada partícula, em cada passo de tempo. Apesar disso, a técnica apresenta bons resultados.

Em uma análise preliminar do estado da arte, nossa revisão sistemática mostrou que o método SPH - de simples implementação e versátil, já testado com uma ampla gama de aplicações; até então foi pouco explorado para a simulação de fluidos biológicos.

1.4 Objetivo

O **objetivo** de nosso trabalho é adaptar uma ferramenta computacional para simulação do sangue, baseado no método SPH; e com isso propor uma melhoria no método com a apresentação de um novo teste de colisão com fronteiras.

Como **objetivos específicos**, elencamos:

1. Adaptação do simulador em SPH para o fluido biológico sangue, incluindo no modelo as especificidades deste fluido e adaptações nas equações do modelo;
2. Adição ao modelo de malhas triangulares representando os vasos sanguíneos;
3. Aplicar correção na densidade, para manter a condição de incompressibilidade;
4. Validar numericamente o simulador, com a comparação de resultados obtidos nas simulações com os calculados por soluções analíticas, que repetem o mesmo escoamento.
5. Aprimorar o teste de colisão implementado, a fim de aumentar a confiabilidade e realismo do simulador;

1.5 Hipóteses e Justificativa

Nossa **hipótese** é que o método SPH pode ser empregado na adaptação de um simulador de sangue, e que aperfeiçoamentos no teste de colisão com fronteira resultarão em um simulador mais confiável.

Como **justificativa**, confirmada nossa hipótese de pesquisa:

- À comunidade científica, estamos colaborando na validação do método SPH para a simulação de fluidos biológicos. Além disso, estamos contribuindo no aperfeiçoamento do método, ao apresentar técnicas como o novo teste de colisão rotacionada.
- Já os profissionais da área da saúde terão à sua disposição um modelo computacional funcional do sangue, adaptável para uso em diversas aplicações (como as listadas na seção 1.1).

1.6 Metodologia

Este trabalho se trata de uma pesquisa prática, de natureza aplicada, exploratória, e de abordagem quantitativa.

As principais atividades que executamos para atingir os objetivos do trabalho foram:

1. Revisão da Literatura acerca da simulação computacional do sangue com uso da técnica SPH, fazendo uso de uma revisão sistemática (BIOLCHINI et al., 2005).

2. Estudo da técnica SPH, sobretudo das equações básicas e parâmetros que caracterizam o modelo.
3. Aprimoramento do simulador em SPH desenvolvido pelo grupo de pesquisa do Prof. Helton para o fluido sangue, levando em consideração suas especificidades e adaptando as equações.
4. Implantação e teste com diferentes vasos sanguíneos, modelados com malhas triangulares.
5. Aprimoramento do modelo para manter a incompressibilidade do sangue.
6. Validação numérica do simulador, através de comparações com escoamentos de solução analítica conhecida - teste de fluxo de Poiseuille.
7. Aprimoramento na técnica de colisão de partículas com as paredes de um vaso sanguíneo, prevenindo a ocorrência de “vazamentos”.

1.7 Organização do trabalho

No capítulo 2, apresentamos a Revisão Sistemática que fizemos a fim de obter o estado da arte sobre o assunto.

No capítulo 3 apresentamos os conceitos fundamentais do método SPH e conteúdos teóricos relacionados, como a Condição CFL e os números quatérnios.

No capítulo 4, exploramos com detalhes a implementação computacional que fizemos, explicitando os parâmetros e funções usadas, e com imagens do simulador funcionando. Também discutimos as implicações que a incompressibilidade do sangue impôs, além de apresentar o método WCSPH.

O capítulo 5 se dedica a explicar o teste de validação numérica que fizemos, do fluxo de Poiseuille, acompanhado de resultados e discussões.

No capítulo 6 temos o Teste de Colisão que propomos, baseado em rotações em 3D com quatérnios, e que resolveu o problema de “vazamento” de partículas que escapavam da malha.

Por fim, no capítulo 7, encerramos o trabalho com as Conclusões finais e proposições de Trabalhos Futuros.

2 Revisão Bibliográfica

Realizamos uma revisão da literatura com o objetivo de obter o atual estado-da-arte da simulação sanguínea baseada no método mesoscópico SPH (Smoothed Particle Hydrodynamics). Além de investigar e compreender os desafios presentes na simulação numérica, pretendemos descobrir quais parâmetros são utilizados e como a simulação pode contribuir com os ambientes virtuais de treinamento médico. (YE; PHAN-THIEN; LIM, 2016).

2.1 Metodologia da Revisão

Para atingir nosso objetivo, usamos uma técnica chamada Revisão Sistemática. Uma Revisão Sistemática (RS) é um método de pesquisa científica planejado, desenvolvido inicialmente para uso em Medicina, que visa responder a uma ou mais questões específicas; e que usa métodos explícitos e sistemáticos para identificar, selecionar e avaliar criticamente os estudos que serão incluídos na revisão (BIOLCHINI et al., 2005). Traz uma série de vantagens, como uma menor chance de conter os vícios de pesquisa do revisor.

Uma RS pode ser resumida em três fases: Planejamento, Execução ou Condução e Análise de Resultados. No planejamento, definimos os objetivos da pesquisa, qual pergunta de pesquisa estamos tentando responder, entre outros conceitos que guiarão todo o trabalho de revisão. Nesta fase, também é preparado o formulário de Protocolo, que resume toda a revisão.

Na fase de Execução/Condução, pesquisamos em repositórios e bancos de dados seguindo as diretrizes descritas no Protocolo.

Por fim, na fase de Análise de Resultados, separamos os resultados relevantes e, a partir deles, extraímos os dados. Numa fase maior (Empacotamento), integrando as 3 fases anteriores, obtemos os resultados esperados quando iniciamos a pesquisa (KITCHENHAM, 2004).

Apresentamos os principais tópicos do Protocolo que desenvolvemos para conduzir nossa revisão, que é uma mistura dos modelos propostos por Biolchini et al. (2005) e Kitchenham (2004):

2.1.1 Protocolo da Revisão Sistemática

Objetivo: Estudar como o método SPH tem sido usado para simulação computacional do fluido sanguíneo; para obter o atual estado da arte.

Questão de pesquisa principal: Como o método SPH foi usada para simular o sangue? Em quais contextos? Quais são os parâmetros? Como é validado? São necessárias mudanças no modelo? As simulações são realizadas em 2D ou 3D? Como o sangue foi parametrizado?

Resultados esperados (finalidade da RS): Conhecimento do uso atual do método SPH para simular o fluxo sanguíneo, identificando particularidades, avanços e pontos que podem ser aperfeiçoados / lacunas.

Palavras-chave utilizadas (keywords): blood, simulation, computational, sph, smoothed particle hydrodynamics.

Idioma dos estudos: Inglês, Português ou Espanhol.

Bases de dados consultadas:

- **Scopus:** Scopus é um banco de dados de resumos e citações de artigos para periódicos acadêmicos. Abrange cerca de 19.500 títulos de mais de 5.000 editoras internacionais, incluindo a cobertura de 16.500 periódicos revisados por pares nas áreas científicas, técnicas, médicas e sociais (incluindo artes e humanidades). É de propriedade da Elsevier e disponibilizado na Web para seus assinantes.
- **IEEE Xplore:** A biblioteca digital IEEE Xplore é um poderoso recurso para a descoberta de conteúdo científico e técnico publicado pelo IEEE (Institute of Electrical and Electronics Engineers) e seus parceiros. O IEEE Xplore fornece acesso pela web à mais de quatro milhões de documentos em texto completo de algumas das publicações mais citadas do mundo em engenharia elétrica, ciência da computação e eletrônica. Está disponível na Web para assinantes

Artigo de Controle: Usamos a revisão (não sistemática) desenvolvida em 2016 por Ye, Phan-Thien e Lim (2016) como referência inicial.

Critérios de inclusão:

- Foco em simulação do sangue.

- Trabalhos recentes (a partir do ano 2000), publicados em revistas ou anais de congressos.
- É usado especificamente o método sem malhas SPH ou algumas de suas derivações.

Critérios de exclusão:

- Trabalhos de simulação não baseados em partículas.
- Obras não disponíveis na versão completa.

Sumarização dos Resultados: Depois de ler e resumir os trabalhos selecionados, elaboramos um relatório técnico com uma análise quantitativa. Uma análise qualitativa também foi desenvolvida para identificar as vantagens e os desafios na utilização deste método.

2.2 Resultados

O período de condução durou 4 meses (de outubro de 2017 a janeiro de 2018). Identificamos, inicialmente, 38 artigos que corresponderam às palavras-chave (mencionadas na seção 2.1.1) utilizadas nas buscas pelas bases de dados. Destes, 29 foram excluídos e 9 foram utilizados, de acordo com os critérios de inclusão e exclusão estabelecidos no Protocolo.

Após a leitura cuidadosa dos 9 artigos selecionados, identificamos as seguintes tendências comuns no estado da arte da simulação computacional do fluxo sanguíneo com o método SPH:

2.2.1 Variedade de aplicações

Como seria de esperar, todos os trabalhos estão de algum modo relacionados com a medicina. Mas mesmo dentro dessa grande área, podemos identificar uma subdivisão (veja Figura 2.2.1).

2.2.2 Colisões em Fronteiras

A interação entre partículas e paredes requer atenção especial para impedir a penetração, e também para modelar corretamente as propriedades do fluido. Uma variedade

Figura 1 – Finalidades dos artigos

Fonte: Thiago Sousa, 2018

de métodos tem sido proposta para evitar que partículas dos fluidos passem através dos limites sólidos - a maioria usa partículas para modelar as paredes de limites, ou usam partículas fantasmas.

Em nossa revisão, o artigo de Kulp et al. (2013) desenvolveu uma técnica que permite que as partículas de contorno mudem naturalmente de pressão com o restante do fluido, o que provou ser eficaz, estável e um método preciso.

Shahriari et al. (2012) analisaram a simulação do sangue em uma cavidade do coração, implementando a condição de contorno da parede com um conjunto de partículas finitas exatamente nos limites (localizadas à mesma distância das partículas iniciais do fluido) e algumas camadas de partículas imaginárias fixas (fantasmas), fora dos limites paralelos das paredes. Essa abordagem foi eficiente por conseguir modelar fluxos altamente inerciais com uma resolução de partículas relativamente baixa.

No trabalho de Holland, Pop e John (2017), a fronteira foi representada como uma amostragem da superfície. Essas partículas da fronteira, então, exercem uma força sobre o fluido, que potencialmente deforma a forma original da malha. Assim, foi possível modelar com precisão as particularidades encontradas no sistema de circulação e trabalhar com uma geometria complexa importada de um scanner médico.

2.2.3 Diferentes maneiras de fazer a Busca por Partículas Vizinhas e Tratar colisões

Para melhorar a eficiência de seus algoritmos, alguns autores alteraram a maneira pela qual a busca por partículas vizinhas é feita e, eventualmente, o teste de detecção de colisão.

No trabalho de Kulp et al. (2013), por exemplo, os autores precisaram mudar o modelo de colisão polígono-esfera para um modelo de colisão esfera-esfera, que pode ser otimizado por GPUs (Unidades de Processamento Gráfico, do inglês *Graphics Processing Units*).

No trabalho de Holland, Pop e John (2017), a busca de partículas vizinhas foi baseada no tradicional sistema de grade uniforme, com a operação de busca sendo realizada na GPU.

A fim de reduzir a complexidade computacional, o trabalho de Shahriari et al. (2012) mudou a busca (baseada em distância, pelo raio do núcleo) para uma forma otimizada, baseada em uma lista encadeada de partículas vizinhas. Isso garantiu a redução da complexidade do algoritmo de busca de partículas de N^2 para $N \log N$.

No trabalho de Xiao et al. (2017), um método de grade espacial foi introduzido para alcançar a busca por partículas vizinhas mais próximas que atravessam as fronteiras. Essa abordagem reduziu a área de pesquisa e também reduziu o número de partículas envolvidas na computação, melhorando assim a eficiência do programa.

Finalmente, Huang et al. (2015), analisando uma mistura de fluidos, trocaram a busca por exaustão por uma pesquisa euleriana em grade uniforme, com comprimento h (h é o suavizador do raio do núcleo). Como resultado, eles aceleraram o processo de pesquisa e usaram um método de hash.

2.2.4 Aceleração

Embora o método SPH seja considerado mais rápido e mais estável para simulação computacional do sangue, em uma escala mesoscópica (YE; PHAN-THIEN; LIM, 2016), alguns dos artigos revisados otimizaram ainda mais o tempo de execução, acelerando partes do código com GPUs (*Graphics Processing Units*).

Quase por unanimidade, esses trabalhos separaram trechos do código em vários *threads* para serem executados em GPUs, que têm uma arquitetura de processamento paralela. Todas as GPUs usadas nos trabalhos pesquisados foram da fabricante Nvidia, que funciona com a linguagem de programação CUDA (*Compute Unified Device Architecture*). Apesar do notório problema de comunicação lento entre GPUs e CPUs, todos os trabalhos em que as GPUs foram utilizados trouxeram melhorias significativas no desempenho do conjunto; como Huang et al. (2015), Shao, Wang e Zhao (2016), Xiao et al. (2017), Holland, Pop e John (2017) e Kulp et al. (2013).

O trabalho cujos resultados apresentaram algumas diferenças em relação aos anteriores foi o de Grønli et al. (2017): o simulador baseado em GPU que eles construíram para obter a evolução de partículas teve um funcionamento “em tempo real” para sistemas 2D; mas em 3D um resultado um pouco mais lento.

2.2.5 Simulações em 2D x 3D

Alguns dos artigos pesquisados fizeram simulações em 2D do fluxo sanguíneo. Nestes trabalhos, a preocupação foi ajustar o SPH ao objeto de estudo, mantendo as duas dimensões (SHAHRIARI et al., 2012; SHI; ZHENG; LIU, 2017; AL-SAAD; KULASEGARAM; BORDAS, 2016). Os autores desses artigos também concordam que, ao adaptarem seus simuladores para funcionarem em 3D, introduziram novas dificuldades.

Nos trabalhos onde a simulação aconteceu em 3D (a maioria), quase sempre foram utilizadas as diretrizes gráficas OpenGL. Há um consenso entre esses autores de que uma animação precisa ter pelo menos 10 fps (frames/quadros por segundo) para ser minimamente convincente, enquanto uma taxa de 30 fps seria a ideal. Todos os trabalhos pesquisados apresentaram taxas entre 10 e 30 fps, sendo que alguns deles também utilizavam técnicas de renderização adicionais para melhorar a qualidade da visualização gerada.

Shao, Wang e Zhao (2016), por exemplo, desenvolveram um novo cálculo de tensão de superfície, que exibia imagens em uma escala de cores.

Xiao et al. (2017) também desenvolveram uma função em escala de cores. Além disso, eles usaram as técnicas de *marching cubes* para triangular a superfície e, em seguida, outra técnica de triangulação que substitui um triângulo plano original por uma forma curva que pode ser re-triangulada em outras unidades menores, gerando imagens 3D realistas e imersivas.

No trabalho de Kulp et al. (2013), uma combinação de métodos foi usada: primeiro um filtro de suavização é aplicado à imagem e, a partir dela, uma malha é desenhada. Uma ferramenta de modelagem 3D limpa a imagem, que passa por uma interpolação *spline cúbica* para chegar ao resultado final, com riqueza de detalhes.

2.2.6 Simulação de Várias Fases

Alguns dos trabalhos estudados simularam o sangue com algum outro fluido, o que requereu alguns ajustes no modelo.

No trabalho de Al-Saad, Kulasegaram e Bordas (2016), por exemplo, o objetivo era estudar a formação de trombose em um determinado paciente. A trombose, como definido no artigo, é o acúmulo de plaquetas (trombos) em um vaso sanguíneo danificado. Assim, apesar de sua simulação focar no sangue, foi necessário modelar partículas com características de plasma (maioria) e outras com características de plaquetas. Além disso, foi necessário modelar um vaso sanguíneo danificado e incluir no modelo forças de adesão e agregação - forças estas exercidas apenas pelas plaquetas. O resultado permitiu assistir a formação do trombo e concluir que, à medida que a velocidade do fluxo sanguíneo aumenta (até um certo limite), a taxa de crescimento da trombose também aumenta.

Huang et al. (2015) simularam a entrada do fluido contraste e sua difusão no fluxo sanguíneo, uma das etapas do exame de *angiografia*; que identifica vasos sanguíneos danificados. Foi necessário modelar as partículas de contraste (que geralmente têm mais viscosidade que o sangue) com base no SPH, para elaborar um modelo de retenção na superfície de sólidos (adsorção) para este contraste, além de desenvolver uma metodologia para interação fluido-fluido e outra para interação fluido-sólido.

No trabalho de Shao, Wang e Zhao (2016), os autores simularam a interação entre o sangue e um tecido mole, como um fígado; um item essencial para o treinamento cirúrgico. O desafio neste trabalho foi unificar toda a simulação para um modelo comum em SPH. As partículas de sangue e tecido mole foram modeladas em conjunto, e foram incluídos recursos que permitiram identificar e diferenciar a natureza das partículas. Com uma adaptação na simulação do sangue, dos tecidos moles e a interação entre eles (que se tornou uma interação fluido-fluido), foi possível obter imagens estáveis e em tempo real.

2.2.7 Interação com dispositivos não convencionais

Apenas dois estudos pesquisados previram interação em tempo real pelos médicos: um para o treinamento cirúrgico, e outro para o planejamento de tratamento médico.

No artigo de Holland, Pop e John (2017), o usuário é capaz de manusear um conjunto de Realidade Virtual (chamado *zSpace*) para visualizar em 3D a anatomia específica dos vasos sanguíneos de um determinado paciente em tempo real. Ele pode usar uma caneta *stylus* para interagir com o modelo e planejar uma intervenção cirúrgica, por exemplo.

O recente trabalho de Xiao et al. (2017) teve como foco o treinamento de médicos para cirurgias minimamente invasivas. Eles desenvolveram um simulador completo de secção de tumor cerebral. Este simulador permite a interação com até duas mãos, usando instrumentos médicos com retorno visual realista - incluindo texturização e iluminação de alta resolução, sombras de moldes suaves, profundidade de campo, distorções de lentes e até sangramentos.

2.2.8 Aquisição de dados, validação e valores específicos

Em geral, os dados usados para modelar o coração, sangue, doenças e outros; vem de dados e exames reais, garantindo simulações com resultados confiáveis e próximos da realidade (SHAHRIARI et al., 2012; AL-SAAD; KULASEGARAM; BORDAS, 2016; KULP et al., 2013; HOLLAND; POP; JOHN, 2017).

A validação de resultados foi um item que nem todos os trabalhos pesquisados previram. Alguns artigos têm conjuntos de dados muito específicos que serviram como *benchmark* para validação do modelo (SHAHRIARI et al., 2012; HOLLAND; POP; JOHN,

2017; GRØNLI et al., 2017). Outros, como Kulp et al. (2013) e Holland, Pop e John (2017) testaram o uso com cirurgias renomadas para validar seus simuladores.

Todos os artigos fornecem pseudo-códigos e explicações das implementações que foram feitas, mas poucos apresentam os dados utilizados; com características e valores numéricos. As exceções foram os trabalhos de Shahriari et al. (2012) e Al-Saad, Kulasegaram e Bordas (2016), que apresentam com mais detalhes os parâmetros utilizados na simulação.

O número de partículas no sangue, por exemplo, a quantia escalar mais mencionada nos trabalhos pesquisados, variou de 1.000 partículas (SHAO; WANG; ZHAO, 2016) até 100^3 partículas (KULP et al., 2013).

2.3 Conclusões

Foi realizado um mapeamento sistemático sobre o uso de SPH na simulação do fluxo nos vasos sanguíneos. Analisamos os trabalhos de acordo com os métodos de colisão em fronteiras, técnicas de busca de vizinhança, aceleração, dimensão da simulação (2D ou 3D), simulação multifásica, interação com dispositivos de entrada e métodos de validação adotados.

O método SPH é promissor. Há alguns anos, fazer uma simulação computacional do fluxo sanguíneo poderia levar até uma semana (KULP et al., 2013), e hoje pode ser feito em tempo real com maior riqueza de detalhes.

O método tem uma ampla gama de uso, e pode ser a base para o treinamento médico de cirurgias minimamente invasivas, sistemas de apoio à decisão para médicos, desenvolvimento de novos equipamentos médicos, investigação de doenças, entre outros.

A pesquisa mostrou que o uso de SPH como método numérico para simular fluidos biológicos ainda é algo novo (poucos trabalhos encontrados, e somente a partir de 2003). Há uma expectativa de aumento no uso desse método, pois ele será a base de alguns sistemas do “Cyber Hospital do Futuro” (HOLLAND; POP; JOHN, 2017).

Uma lacuna identificada em nosso estudo é a falta de ambientes de simulação que permitam a interação com o usuário através de dispositivos não convencionais, como os hápticos (sensíveis ao tato).

Ainda há espaço para aumentar o realismo e as possibilidades com esse tipo de simulação, especialmente quando se trata de aplicativos que permitem interação em tempo

real com os usuários - sejam eles simuladores de cirurgia virtual ou sistemas de auxílio de diagnóstico. No entanto, o número de pesquisadores dedicados a essa temática está aumentando, o que nos faz prever que sistemas como esses se tornarão mais comuns em um futuro não muito distante.

3 Conceitos Fundamentais

Na dinâmica dos fluidos, o movimento e o transporte de energia e substância são governados por leis físicas, como a da conservação da massa, da quantidade de movimento, da energia e da substância, as quais podem ser escritas matematicamente por equações diferenciais. Essas equações são frutos dos trabalhos realizados pelo francês Claude Navier (1785-1836), descrita na obra “*Mémoire sur les lois du mouvement des fluides*”, no ano de 1822; posteriormente retrabalhadas para a forma como são conhecidos na literatura atual, pelo irlandês George Gabriel Stokes (1819-1903), em seu artigo “*On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solid*”, datado de 1845 (PINTO, 2013). Essas equações são apresentadas na literatura especializada como as **equações de Navier-Stokes**. Tradicionalmente, as equações de Navier-Stokes são apresentadas na forma euleriana (equação da continuidade (1) e equação do momento (2)):

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \quad (1)$$

$$\frac{\partial \mathbf{v}}{\partial t} - \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mathbf{S} + \mathbf{g} \quad (2)$$

Onde t representa o tempo, $\mathbf{v}$ o vetor de velocidade, ρ a densidade, p a pressão, $\mathbf{g}$ o vetor de aceleração da gravidade, $\mathbf{S}$ o tensor extra-tensão e ∇ o operador de gradiente.

O tensor $\mathbf{S}$ é calculado a partir do coeficiente de viscosidade μ do fluido e pelo tensor taxa de deformação $\mathbf{D}$:

$$\mathbf{S} = 2\mu \left(\mathbf{D} - \frac{1}{3} \overline{(\mathbf{D})} \mathbf{I} \right) \quad (3)$$

e o tensor $\mathbf{D}$ calculado segundo a equação 4:

$$\mathbf{D} = \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \quad (4)$$

onde $\nabla \mathbf{v}^T$ é a transposta de $\nabla \mathbf{v}$, e $\mathbf{I}$ o tensor identidade. Observe também que $\overline{(\mathbf{D})} = \nabla \cdot \mathbf{v}$.

O termo viscoso $\nabla \cdot \mathbf{S}$ pode conter várias variáveis para serem calculadas, de forma que seu uso não pode ser generalizado: deve-se especificar o tensor $\mathbf{S}$ de acordo com o comportamento do fluido. Caso o fluido seja incompressível, e possua viscosidade μ constante, o termo $\nabla \cdot \mathbf{S}$ pode ser reescrito na forma $\mu \nabla^2 \mathbf{v}$.

As equações de Navier-Stokes possuem um alto grau de dificuldade para serem solucionadas analiticamente, sendo necessário o desenvolvimento de métodos numéricos - que auxiliam na resolução de equações mais complexas. Neste contexto, se desenvolve o campo de estudos da Dinâmica dos Fluidos Computacional (do inglês, *Computational Fluid Dynamics* – CFD), desenvolvendo uma nova forma de analisar o comportamento dos fluidos, com o apoio de computadores.

Na dinâmica dos fluidos computacional, existem dois tipos de abordagens, se diferenciando nas formas de resolução:

- os eulerianos, fundamentado em malhas (*Grid-based Methods*);
- os lagrangianos, sem malhas (*Particle-based Methods*).

Contudo, nem todos os métodos obtêm resultados precisos e satisfatórios, devido ao alto grau de complexidade das equações e dos métodos numéricos que são desenvolvidos na simulação, sejam eles eulerianos ou lagrangianos.

Os métodos numéricos de natureza euleriana necessitam de malhas. Os mais clássicos são: os das diferenças finitas, dos elementos finitos e os dos volumes finitos. Entretanto, esses métodos possuem problemas com perfis específicos, que não são resolvidos eficientemente pelos métodos com malhas, tais como os que possuem geometrias complexas, geometrias móveis ao longo do tempo, descontinuidades do domínio, grandes deformações, deformação localizada e com necessidade de refinamento, o que torna a solução mais dispendiosa, do ponto de vista computacional (GUEDES, 2006).

Entre os métodos numéricos lagrangianos, no contexto da dinâmica dos fluidos computacional, existem diversos métodos que não usam malhas, como: Smoothed Particle Hydrodynamics (SPH), Diffuse Element Method (DEM), Element Free Galerkin Method (EFGM) e o Meshless Local Petrov-Galerkin (MLPG) (VASCO, 2014).

O método SPH foi criado na década de 1970 por Gingold e Monaghan (1977) e Lucy (1977), em um contexto de simulação de fluidos incompressíveis em astrofísica (estudo de colisões entre as estrelas, formação de galáxias e supernovas) com uma formulação mais simples e baseada em partículas. Substitui-se o fluido por um conjunto finito de partículas, e depois é desenvolvida uma interpolação de dados.

Para calcular os valores das grandezas físicas de interesse são utilizadas as “funções (ou *kernels*) de suavização”, interpolando os valores existentes dentro de uma vizinhança fechada (determinada por um raio de suporte compacto, que se localiza em torno de um

ponto material/partícula de interesse). Ou seja, o estado de cada partícula dependerá do estado que se encontram as “partículas vizinhas” (LIU; LIU, 2003).

O significado da sigla SPH exprime a filosofia do método: o primeiro termo, *Smoothed*, representa a suavização ou medianização segundo o peso das variáveis no campo da vizinhança da partícula, visando principalmente a estabilidade do método. O termo *Particle* refere-se à utilização de partículas para discretização do domínio físico. O terceiro termo, *Hydrodynamics*, representa a área de atuação deste método. Assim, a combinação da adaptabilidade, natureza particulada e lagrangiana leva o método SPH a ser aplicado em diferentes áreas da engenharia e ciência. De certa maneira, o termo *Hydrodynamics* pode ser interpretado como mecânica dos fluidos, de forma geral. Daí o nome *Smoothed Particle Hydrodynamics* (VASCO, 2014).

O Método SPH foi aplicado em uma gama versátil de simulações, como em problemas envolvendo escoamento de fluidos compressíveis, modelagens de materiais deformáveis, e também de materiais líquidos: lavas vulcânicas, lama e óleo (MILLER; PEARCE, 1989).

3.1 Fundamentos do Método

Nesse método, a amostra é representada por um conjunto finito de partículas que possuem propriedades próprias, individuais, que se movem de acordo com as equações de Navier-Stokes. As partículas se encontram dentro de um volume determinado por um raio de suporte, que contribuem para o comportamento da partícula pontuada naquele instante, a partícula i , também presente no fluido (ver Figura 2).

O método SPH consiste num método de interpolação das propriedades físicas de um sistema hidrodinâmico e de aproximação das derivadas espaciais de campos contínuos utilizando um conjunto discreto de partículas. A principal ideia do método SPH é baseada no conceito da representação integral de uma função contínua $f : \mathbb{R}^d \rightarrow \mathbb{R}$ definida num domínio aberto $\Omega \subset \mathbb{R}^d$, tal que $\mathbf{x} \in \Omega$ e que satisfaz a identidade:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (5)$$

Onde $\mathbf{x}$ é um vetor d -dimensional e $\delta(\mathbf{x} - \mathbf{x}')$ é a função Delta de Dirac, que por definição deve satisfazer:

Figura 2 – Significado da Função Núcleo. Na figura (a) vemos as aproximações por partículas, usando partículas dentro do raio suporte da função de suavização W para a partícula i . A busca por vizinhança possui formato circular com raio kh . Na figura (b), as alturas das colunas ilustram o “peso” que cada partícula exerce na partícula de interesse (inversamente proporcional à distância).

Fonte: adaptado de Vasco (2014) e Liu e Liu (2003)

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} 1, & \text{se } \mathbf{x} = \mathbf{x}' \\ 0, & \text{se } \mathbf{x} \neq \mathbf{x}' \end{cases} \quad (6)$$

Embora não conseguimos definir uma função que satisfaça as propriedades da distribuição Delta de Dirac, a Análise Funcional nos permite definir a distribuição Delta de Dirac como sendo o limite de funções suaves (PAIVA et al., 2009). Portanto, no SPH, podemos aproximar a função $f(\mathbf{x})$ simplesmente substituindo a distribuição Delta de Dirac na equação 5 por uma função suave W . Logo, a equação será reescrita da seguinte forma:

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (7)$$

Onde h , chamado de comprimento suave, define o raio de influência kh da função W , sendo k o fator de escala. A função W é chamado de função kernel de suavização, função núcleo, função de suavização, função kernel, ou simplesmente kernel (LIU; LIU, 2003). A variável h é conhecida como o comprimento do raio de suavização, que define o setor de influência, a uma distância h a partir do ponto x . Usamos $\langle \rangle$ na equação 7 para indicar que o valor de f é aproximado, quando trocamos a função δ pela função W .

Importante observar que W não é a própria função Delta de Dirac, apenas uma função de ponderação que permite interpolar os valores de $f(x)$.

A função de suavização W é escolhida por satisfazer as seguintes propriedades:

1. O kernel deve ser **suficientemente suave**:

$$W \in C^k, k > 1 \quad (8)$$

2. O kernel deve ser **normalizado**:

$$\int_{\mathbb{R}^d} W(x_i - x_j, h) d^n x = 1 \quad (9)$$

3. O kernel deve ter **suporte compacto**:

$$W(x_i - x_j, h) = 0, \text{ quando } |x_i - x_j| \geq kh \quad (10)$$

4. O kernel deve ser **positivo**:

$$W(x_i - x_j, h) \geq 0 \quad (11)$$

5. O kernel deve ser **decrecente**:

$$W(x_i - x_j, h) < W(x_i - x_k, h), \text{ se } |x_i - x_j| < |x_i - x_k| \quad (12)$$

6. O kernel deve ser **simétrico**:

$$W(x_i - x_j, h) = W(x_j - x_i, h) = W(\mathbf{x}) \quad (13)$$

7. O kernel deve satisfazer a **função Delta de Dirac**:

$$\lim_{h \rightarrow 0} W(x_i - x_j, h) = \delta(x_i - x_j, h) \quad (14)$$

A **primeira propriedade** informa que o kernel deve ser suficientemente suave. Para que essa função kernel seja classificada como suave, deve possuir derivadas de todas as ordens (STEWART, 2006).

A **segunda propriedade** informa que um kernel deve ser normalizado, condição da unicidade. Isso garante que a integral da função de suavização, que estará sob o raio suporte, será unitária, garantindo que a interpolação seja correta (GINGOLD; MONAGHAN, 1977).

A **terceira propriedade** afirma que o kernel deve possuir suporte compacto. Essa propriedade tem a finalidade de transformar uma operação global da aproximação do

método SPH, em uma operação local. Se a partícula vizinha estiver acima do raio de procura kh , a função sempre valerá zero.

Esse valor de h é conhecido como raio de suporte do kernel (ou da função de suavização), e tem a função de limitar o número de partículas em sua vizinhança, que, para a ação computacional, é de suma importância. As propriedades físicas da partícula i não dependerão de todas as partículas presentes no sistema, e sim de um grupo delas, as quais são justamente limitadas pelo raio suporte. Se uma função é classificada como compacta, ela é zero fora daquele “conjunto compacto”.

Se houver um h muito pequeno, pode não existir partículas suficientes naquele domínio para que exerça forças sobre determinadas partículas, prejudicando a acurácia da solução. E se houver um h muito grande, podemos ter um maior esforço computacional e também uma perda de acurácia (ALVES, 2008).

A **quarta propriedade** informa que o kernel deve ser não-negativo no suporte. Isso possui grande importância para a interpolação, pois assegura que as aproximações sejam coerentes para as propriedades físicas envolvidas no problema e para a estabilidade. A estabilidade está relacionada à propagação de erros de arredondamento, nos passos intermediários, e pode ser grave nos casos de “métodos multipasso” (PAIVA et al., 2009).

O termo “estabilidade” diz respeito à influência face aos erros iniciais. Quando utilizados valores negativos para o núcleo, temos como consequência valores negativos para a massa específica e para a energia, por exemplo, sendo um absurdo físico (LIU; LIU, 2003).

A **quinta** e a **sexta propriedades**, decrescente e simétrica (função par), são importantes para determinar a influência (em relação ao kernel) das interações entre as partículas, de acordo com as distâncias entre elas. Partículas mais próximas devem influenciar mais do que partículas mais distantes, e partículas nas mesmas distâncias, contudo em posições diferentes, devem influenciar da mesma forma.

A **sétima propriedade** informa que a função de suavização deve satisfazer determinadas condições da função Delta de Dirac. Ou seja, para um h que tende a zero, a função W se aproxima da função δ .

3.2 Aproximação por partículas

O processo de discretização do domínio contínuo do fluido consiste em substituir a integral do domínio Ω pelo somatório realizado sobre as partículas localizadas na vizinhança, essas localizadas sob a influencia do raio suporte h da função de suavização, sendo uma aproximação por partículas materiais. Neste processo o volume infinitesimal dx' , anteriormente descrito na integração da localização da partícula vizinha j , é substituído pelo volume finito de partículas ΔV_j que estão relacionadas com a massa das partículas m_j , e dado por:

$$\Delta V_j = \frac{\rho_j}{m_j} \quad (15)$$

Onde m_j é a massa específica das partículas j , ρ_j é a densidade das partículas j , $j = 1, 2, 3, \dots, N$, N é o número de partículas localizadas dentro do domínio de suporte da partícula j .

A representação da integral contínua do método SPH para função $f(x)$, para a discretização por aproximação por partículas, é escrita da seguinte forma:

$$f(x) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) W_{ij} \quad (16)$$

Onde

$$W_{ij} = W(x_i - x_j, h) \quad (17)$$

Para as derivadas espaciais, temos os seguintes cálculos:

$$\nabla f(x) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) \nabla W_{ij} \quad (18)$$

$$\nabla^2 f(x) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) \nabla^2 W_{ij} \quad (19)$$

Atualmente, as formas mais aceitas e utilizadas para o calculo das derivadas espaciais são (BÍSCARO, 2018):

$$\nabla f(x) = \rho_i \sum_{j=1}^N m_j \left(\frac{f(x_i)}{\rho_i^2} + \frac{f(x_j)}{\rho_j^2} \right) \nabla W_{ij} \quad (20)$$

$$\nabla \cdot f(x) = -\frac{1}{\rho_i} \sum_{j=1}^N m_j f(x_{ij}) \cdot \nabla W_{ij} \quad (21)$$

$$\nabla^2 A_i = 2 \sum_{j=1}^k \frac{m_j}{\rho_j} f(x_{ij}) \frac{x_{ij} \nabla W_{ij}}{x_{ij} \cdot x_{ij} + 0.01 h^2} \quad (22)$$

Onde $f(x_{ij}) = f(x_i) - f(x_j)$, $x_{ij} = x_i - x_j$.

A maior vantagem nessa forma de cálculo (equações 20, 21 e 22 sobre as anteriores (equações 18 e 19) é a simetria em relação aos índices i e j . As equações 20 e 22 são utilizadas no cálculo da aceleração das partículas, já a equação 21 pode ser usada para prever mudanças na densidade a partir do divergente da velocidade.

3.3 Condição CFL

A condição de Courant-Friedrichs-Lewy (CFL) é um requisito necessário para a convergência de alguns métodos numéricos, que garante a estabilidade e convergência na resolução de equações diferenciais parciais.

Este número adimensional recebeu o seu nome devido o artigo produzido pelos cientistas Courant, Friedrichs e Lewy (1928).

Define-se o número de Courant (Co) como

$$Co = \frac{|U| \cdot \Delta t}{\Delta x} \quad (23)$$

onde Co é o número de Courant, $|U|$ é a norma da magnitude de velocidade local, Δt é o intervalo de tempo e Δx é o tamanho local dos volumes finitos da malha.

Co representa, então, o fluxo advectivo que atravessa os volumes da malha em um dado intervalo de tempo. Se supormos uma malha e uma velocidade uniformes numa região do domínio, o Número de Courant local torna-se o número de volumes da malha atravessados por uma perturbação naquele passo de tempo.

Para um cálculo a massa específica constante, a grandeza que sofre advecção é simplesmente a magnitude da velocidade. Por outro lado, se o cálculo acontece em massa específica variável, a advecção de pressão também é importante e a velocidade c do som influencia no transporte dessas perturbações de maneira que Co se torna

$$Co_{\text{massa variável}} = \frac{(|U| + c) \Delta t}{\Delta x} \quad (24)$$

Na resolução de uma equação diferencial parcial, o que torna o passo de tempo físico diferente do numérico está na capacidade de calcular um problema transitório corretamente. A precisão é sempre exigida, isto é, de nada serve um cálculo estável que não representa a física transiente.

Um cálculo transiente explícito (sem iterações no espaço) exige que o número de Courant de cada volume da malha seja inferior à unidade em todo o domínio. Esta condição de que o número de Courant seja inferior a 1 é o que chamamos de Condição CFL.

O número de Courant pode ser controlado alterando o intervalo de tempo ou o espaço. Quando a partição do espaço é reduzida, ou o intervalo de tempo é aumentado, o número de Courant aumenta. Como o número de Courant rege a estabilidade de um problema em um método explícito, é muito importante que se escolha os intervalos com precisão.

3.4 Números Quatérnios

O conjunto dos quatérnios ($\mathbb{H}$), formulados por Hamilton, são uma extensão do conjunto dos números complexos ($\mathbb{C}$).

Eles foram concebidos para serem uma representação dos complexos no espaço 3D (NEVES, 2008), mas resultaram na criação de um número de 4 dimensões (por isso o nome quatérnios).

Um quatérnio tem a seguinte forma:

$$q = q_0 + q_1 \cdot \mathbf{i} + q_2 \cdot \mathbf{j} + q_3 \cdot \mathbf{k}$$

onde $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ são unidades imaginárias, e os coeficientes q_0 , q_1 , q_2 e q_3 pertencem aos reais.

De forma análoga ao que acontece no conjunto dos números complexos, temos que

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$$

Porém, ao contrário dos complexos, a multiplicação nos quatérnios não é comutativa. Para as unidades imaginárias, vale que:

$$\mathbf{ij} = \mathbf{k}$$

$$\mathbf{ji} = -\mathbf{k}$$

Podemos também representar um quatérnio de forma mais condensada, como $q = (s, \vec{v})$. Nessa representação s é um escalar que representa sua parte real, e $\vec{v} = (v_x, v_y, v_z)$ é um vetor de três componentes que representa sua parte imaginária.

A multiplicação dos escalares $q_1 = (s_1, \vec{v}_1)$ por $q_2 = (s_2, \vec{v}_2)$ pode ser expressa em termos de produtos escalares e vetoriais:

$$q_1 q_2 = (s_1 s_2 - \vec{v}_1 \cdot \vec{v}_2, s_1 \vec{v}_2 + s_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (25)$$

Sugerimos a leitura de GUIDORIZZI (2001) para a demonstração da fórmula 25.

O conjugado de um quatérnio, representado por $\bar{q}$ é definido como

$$\bar{q} = (s, -\vec{v})$$

O inverso de um quatérnio q , representado por q^{-1} é tal que

$$q q^{-1} = q^{-1} q = 1$$

E a magnitude de um quatérnio pode ser calculada como o produto do quatérnio com o seu conjugado:

$$|q|^2 = q \bar{q} = s^2 + |\vec{v}|^2$$

3.4.1 Rotações em 3D com Quatérnios

Seja $\vec{r} = (r_x, r_y, r_z)$ um ponto que queremos rotacionar.

Representando esse ponto pelo quatérnio $p = (0, \vec{r})$, com parte real nula, conseguimos usar um quatérnio unitário $q = (s, \vec{v})$ para fazer a rotação R (NEVES, 2008):

$$R_q(p) = q p \bar{q} \quad (26)$$

Como o quatérnio q é unitário, podemos reescrevê-lo como:

$$q = (s, \vec{v}) = \left(\cos \frac{\theta}{2}, \sin \frac{\theta}{2} \vec{n} \right)$$

O vetor unitário $\vec{n}$ é o eixo de rotação, e θ o ângulo (anti-horário) de rotação.

A equação 26 resulta em um novo quatérnio, cuja parte real será zero e a parte imaginária conterá o resultado da rotação.

Se q_1 e q_2 são quatérnios que representam duas rotações distintas, e se $q_3 = q_2q_1$, podemos demonstrar que

$$R_{q_2}(R_{q_1}(p)) = R_{q_3}(p) \quad (27)$$

ou seja, com os quatérnios podemos facilmente realizar composição de rotações.

4 Aspectos Computacionais

A aproximação da densidade é muito importante no método SPH, já que ela basicamente determina a distribuição das partículas e a evolução do método de suavização dessas partículas.

Substituindo a função $f(x)$ na equação 16 por uma função de densidade ρ , obtemos a aproximação em SPH para a densidade (PAIVA et al., 2009):

$$\rho_i = \sum_{j=1}^N m_j W_{ij} \quad (28)$$

Por essa equação, observa-se que a densidade específica de uma partícula i é uma média ponderada de todas as “massas específicas” das partículas vizinhas no domínio do raio suporte.

4.1 Formulação Lagrangeana das Equações de Navier-Stokes

A formulação lagrangeana das equações de Navier–Stokes descreve o comportamento de um fluido no ponto de vista de uma partícula que se move junto com o fluxo do fluido. Definimos a *derivada material* como

$$\frac{d(\ )}{dt} \equiv \frac{\partial(\ )}{\partial t} + \mathbf{v} \cdot \nabla(\ ) \quad (29)$$

Fisicamente, a derivada material representa a variação instantânea de uma grandeza física em relação ao tempo de um elemento de fluido que se move no espaço. Substituindo no lado esquerdo das equações 1 e 2 pela derivada material, as equações de Navier-Stokes na forma lagrangeana são dadas como:

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \quad (30)$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mathbf{S} + \mathbf{g} \quad (31)$$

A equação 31 ainda pode ser simplificada, e fica dessa forma:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{v} + \mathbf{g} \quad (32)$$

4.2 Cálculo da pressão

Para calcular a massa m_i de uma partícula, que é fixa em todas as partículas, usaremos a definição apresentada por Schechter e Bridson (2012), que estabelece

$$m_i = \frac{2}{3}h^3\rho_0 \quad (33)$$

h é o espaço entre partículas e ρ_0 é a densidade de referência do fluido.

A pressão p_i é calculada a partir da densidade ρ_i de cada partícula e da densidade de referência do fluido (ρ_0) segundo a equação de Tait (MONAGHAN, 1994):

$$p_i = B \left(\left(\frac{\rho_i}{\rho_0} \right)^\lambda - 1 \right) \quad (34)$$

Onde ρ_0 é a densidade do fluido em repouso, B e λ são parâmetros relacionados às flutuações do fluido. Monaghan (1994) adotou $\lambda = 7$

O parâmetro B pode ser estimado por:

$$B = \frac{c^2\rho_0}{\lambda} \quad (35)$$

Onde c é velocidade do som no meio. Nas simulações de SPH, c é estimada como sendo aproximadamente dez vezes o módulo da maior velocidade esperada no escoamento.

4.3 Integração temporal

O método SPH permite calcular as derivadas a partir de partículas (amostras) distribuídas aleatoriamente no espaço. Contudo, a equação de Navier-Stokes do momento (equação 1) envolve derivadas espaciais e temporais.

Dentre os diferentes métodos de integração de equações diferenciais ordinárias, existe a necessidade de saber como são calculadas as variáveis desconhecidas. Os esquemas de integração podem ser classificados em métodos explícitos e métodos implícitos. Nos explícitos, o cálculo é direto - ou seja, quando existe apenas uma incógnita a resolver naquele instante atual. Já no implícito, o instante futuro figura como incógnita e é isolado o instante atual (VASCO, 2014).

Como vantagem, os métodos explícitos podem ser implantados de forma mais fácil, além de realizar os cálculos de maneira rápida e eficiente. Todavia a sua estabilidade

depende de como é feita a escolha do passo de tempo, Δt , para que haja convergência. O método explícito mais simples, neste caso, é o Método de Euler: essa estimativa de derivada envolve apenas dois pontos de tempo: t e Δt , o que permite calcular o instante a seguir diretamente a partir do instante atual, tornando o integrador eficiente - pois basta calcular e armazenar apenas um instante para calcular o seguinte. A velocidade $\mathbf{v}_i$ e a posição $\mathbf{x}_i$ são atualizados da seguinte maneira:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \mathbf{a}_i(t) \quad (36)$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t) \quad (37)$$

onde $\mathbf{a}_i(t)$ é a aceleração da partícula num instante de tempo t .

Todavia o Método de Euler possui uma precisão de primeira ordem e pode ser tornar instável, fato indesejável para qualquer integrador.

Tão simples quanto o método de Euler, o método *Leap-frog* se torna mais adequado, devido a sua precisão de **segunda ordem**. Na prática, este método se torna mais usado porque requer pouca exigência de armazenamento de memória, sem perder a sua eficiência (LIU; LIU, 2003). A base deste método está em calcular as posições e velocidades em instantes de tempo intercalados e dispostos de uma determinada maneira (ver Figura 3). Como exemplo, a posição é calculada em um instante de tempo inteiro, e a velocidade em cada “instante e meio” (ALVES, 2008). Ele é baseado na estimativa de derivada como diferença centrada:

$$\frac{dv}{dt} = \lim_{h \rightarrow 0} \left(\frac{v\left(t + \frac{1}{2}h\right) - v\left(t - \frac{1}{2}h\right)}{h} \right) \approx \frac{v\left(t + \frac{1}{2}\Delta t\right) - v\left(t - \frac{1}{2}\Delta t\right)}{\Delta t} \quad (38)$$

A estimativa da derivada envolve novamente apenas dois pontos de tempo: $\left(t - \frac{1}{2}\Delta t\right)$ e $\left(t + \frac{1}{2}\Delta t\right)$. Porém, para calcular o instante a seguir a partir do instante atual, usa-se instantes intermediários, o que não cria um problema no ponto de vista da aproximação. Na prática, a velocidade de cada partícula é calculada nos pontos médios dos intervalos de tempo, e a posição, nos pontos inteiros.

Figura 3 – Ilustração do esquema de integração numérica Leap-frog.

Fonte: adaptado de Neto (2007)

Nesse esquema de integração a velocidade de cada partícula é calculada nos pontos médios dos intervalos de tempo. Assim, as posições e as velocidades são avaliadas das seguintes formas:

$$\mathbf{v}_i \left(t + \frac{1}{2} \Delta t \right) = \mathbf{v}_i \left(t - \frac{1}{2} \Delta t \right) + \Delta t \mathbf{a}_i(t) \quad (39)$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i \left(t + \frac{1}{2} \Delta t \right) \quad (40)$$

Quando é necessário ser calculada a velocidade em um passo inteiro, ela pode ser interpolada a partir dos passos intermediários. O mais prático é realizar tal calculo através da média entre as velocidades anterior e posterior:

$$\mathbf{v}_i(t) = \frac{1}{2} \left[\mathbf{v}_i \left(t + \frac{1}{2} \Delta t \right) + \mathbf{v}_i \left(t - \frac{1}{2} \Delta t \right) \right] \quad (41)$$

A iniciação do método requer uma etapa adicional no cálculo da velocidade $\mathbf{v}_i \left(-\frac{1}{2} \right)$ através do método de Euler:

$$\mathbf{v}_i \left(-\frac{1}{2} \right) = \mathbf{v}_i(0) - \frac{1}{2} \Delta t \mathbf{a}_i(0) \quad (42)$$

4.4 Funções núcleo

As funções de núcleo podem melhorar significativamente a precisão do método SPH, e ter impacto na estabilidade e eficiência computacional. Procuramos usar funções núcleo estáveis, com acurácia e rapidez de cálculo. As funções utilizadas neste trabalho foram extraídas do trabalho de Müller, Charypar e Gross (2003). A função principal é:

$$W_{poly6}(h, r) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^2, & \text{se } 0 \leq r \leq h \\ 0, & \text{caso contrário} \end{cases} \quad (43)$$

onde $r = |x_i - x_j|$ e h é o raio de influência da partícula.

Essa função núcleo consegue calcular a maioria das grandezas físicas. Entretanto, se essa função for utilizada no cálculo das forças de pressão, as partículas tenderão a formarem agrupamentos sob alta pressão e, além disso, as forças de repulsão tenderiam a desaparecer - uma vez que o seu gradiente se aproxima de zero no centro, o que colocaria as partículas muito próximas umas das outras. Uma maneira de contornar esse problema é utilizar uma função núcleo especialmente projetada para que seu gradiente não se anule no centro. A função utilizada foi a desenvolvida por Desbrun e Gascuel (1996):

$$W_{spiky}(h, r) = \frac{15}{\pi h^3} \begin{cases} (h - r)^3, & \text{se } 0 \leq r \leq h \\ 0, & \text{caso contrário} \end{cases} \quad (44)$$

O gradiente desta função é calculado como:

$$\nabla W_{spiky}(h, r) = \frac{45}{\pi h^6} \begin{cases} (h - r)^2 \hat{r}, & \text{se } 0 \leq r \leq h \\ 0, & \text{caso contrário} \end{cases} \quad (45)$$

Já para o cálculo da viscosidade¹, que é um fenômeno causado pela fricção (que reduz a energia cinética) precisamos de uma função com efeito suavizador no campo de velocidades. As funções núcleo anteriores (POLY6 e SPIKY) tem o operador Laplaciano negativo em um determinado intervalo, o que pode fazer com que a velocidade das partículas aumente - alterando assim a propriedade de viscosidade. Müller, Charypar e Gross (2003) desenvolveram a seguinte função núcleo que resolve esse problema:

$$W_{viscosity}(h, r) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & \text{se } 0 \leq r \leq h \\ 0, & \text{caso contrário} \end{cases} \quad (46)$$

Seu operador Laplaciano é:

$$\nabla^2 W_{viscosity}(h, r) = \frac{45}{\pi h^6} (h - r) \quad (47)$$

---

¹ é a propriedade física que caracteriza a resistência de um fluido ao escoamento, isto é, ao transporte microscópico de quantidade de movimento por difusão molecular. Ou seja, quanto maior a viscosidade, menor será a velocidade com que o fluido se movimenta.

Este operador Laplaciano satisfaz o requisito de ser não-negativo, e assim pode garantir as propriedades da força de viscosidade (SHI; ZHENG; LIU, 2017).

4.5 Estabilidade numérica

Simulações computacionais que objetivam a visualização, como a do sangue, necessitam ser estáveis. O passo de tempo deve respeitar uma condição conhecida como CFL (*Courant–Friedrichs–Lewy*) para garantir essa estabilidade (NETO, 2007):

$$\Delta t = 0,1 \min \left\{ \frac{h}{\|v_i\| + c} \right\} \quad (48)$$

onde c é a velocidade do som no meio. Os autores Dagenais, Gagnon e Paquette (2012) discordam, pois uma pressão muito forte por consequência pode acarretar uma aceleração mais forte do que c , comprometendo a estabilidade do método. Além disso, quando o fluido está em repouso, o impacto da pressão sobre as forças geradas pela velocidade é menor do que c , o que não resultaria num passo de tempo adequado. Ao utilizar a aceleração máxima ao invés de c , Dagenais, Gagnon e Paquette (2012) apresentam um modelo para um passo de tempo maior, e que ainda assim mantém a estabilidade da simulação:

$$\Delta t = - \frac{\|v_{max}^{(t-\Delta t)}\| - \sqrt{\|v_{max}^{(t-\Delta t)}\|^2 + 4 \|a_{max}\| \beta h}}{2 \|a_{max}\|} \quad (49)$$

onde a_{max} representa a máxima aceleração calculada no passo de tempo corrente, e v_{max} a velocidade máxima.

4.6 Suavização da Velocidade XSPH

Com o objetivo de prevenir a interpenetração de partículas, o que pode gerar aglomerados instáveis de partículas, Monaghan (1989) utilizou a técnica *eXtend Smoothed Particles Hydrodynamics* (XSPH) na simulação de escoamentos de fluidos incompressíveis envolvendo superfícies livres e com altas velocidades, de modo que as partículas se movimentem de uma forma mais suave e reduzindo assim o problema da desordem de partículas.

A técnica consiste em calcular uma média, ponderada pela função de suavização, das velocidades das partículas vizinhas e corrigir a velocidade de cada partícula através desse valor médio, multiplicado por um parâmetro constante $\epsilon \in [0, 1]$:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \epsilon \sum_{j=1}^N \frac{2m_j}{\rho_i + \rho_j} (\mathbf{v}_i - \mathbf{v}_j) W_{ij} \quad (50)$$

assim, o deslocamento das partículas é calculado usando essa velocidade corrigida.

4.7 Algoritmo geral

A seguir, descrevemos o pseudo-código (Algoritmo 1) do nosso simulador SPH:

Algoritmo 1 Algoritmo básico do método SPH

- 1: **para** cada partícula $i \leftarrow 1$ **até** N **faça** ▷ Atualização da vizinhança
  - 2:   Encontre os k vizinhos mais próximos de i ;
  - 3: **fim para**
  - 4: **para** cada partícula $i \leftarrow 1$ **até** N **faça** ▷ Atualização de densidade e pressão
  - 5:   Atualize a densidade segundo a fórmula $\rho_i = \sum_{j=1}^k m_j W_{ij}$ (Equação 28);
  - 6:   Calcule a pressão p_i usando a densidade ρ_i (Equação 34);
  - 7: **fim para**
  - 8: **para** cada partícula $i \leftarrow 1$ **até** N **faça** ▷ Atualização das forças atuantes
  - 9:   Calcule $F_i^{pressao} = -\frac{m_i}{\rho_i} \nabla p_i$ (Equação 20);
  - 10:   Calcule $F_i^{viscosidade} = -m_i \mathbf{v} \nabla^2 v_i$ (Equação 22);
  - 11:   Calcule $F_i^{externas} = -m_i g$;
  - 12:   Calcule $F_i(t) = F_i^{pressao} + F_i^{viscosidade} + F_i^{externas}$;
  - 13: **fim para**
  - 14: **para** cada partícula $i \leftarrow 1$ **até** N **faça** ▷ Integração temporal
  - 15:   Atualize a velocidade $v_i(t + \Delta t)$;
  - 16:   Atualize a posição $x_i(t + \Delta t)$;
  - 17:   Realize o Teste de Colisão ▷ Algoritmo 2
  - 18: **fim para**
-

Fonte: Adaptado de Neto (2007)

O primeiro laço do algoritmo 1 (linha 1) executa uma das mais importantes tarefas numa simulação de SPH, que é a atualização das relações de vizinhanças das partículas. A maioria dos métodos descritos na literatura se dividem em duas abordagens para este problema: estruturas hierárquicas, ou grades regulares. O trabalho de Neto (2007) recomenda que se use pelo menos 56 vizinhos por partícula em simulações 3D.

O segundo laço, que se inicia na linha 4 do algoritmo 1 realiza a atualização da densidade e da pressão em cada partícula a partir da densidade, o que é conhecido na literatura como equação de estado (equação 34).

Já o terceiro laço, iniciado na linha 8 do mesmo algoritmo atualiza todas as componentes das forças atuantes em cada partícula. A força resultante no instante t , $F_i(t)$ será a aceleração incidente em cada partícula e será utilizada para o último passo de integração no tempo.

O último laço, iniciado na linha 14, atualiza a posição, velocidade, e outras propriedades das partículas.

4.8 Simulador SPH

O simulador que utilizamos foi desenvolvido pelo Prof. Dr. Helton H. Bísaro (2018).

Ele foi construído em C++, compilado com as diretivas CMake² e usa os pacotes gráficos OpenGL e CGAL (Computational Geometry Algorithms Library).

O **OpenGL**³ (**Open Graphics Library**) é uma API (Application Programming Interface) livre, desenvolvida pelo Khronos Group e utilizada na computação gráfica para desenvolvimento de aplicativos gráficos, ambientes 3D, jogos, entre outros. OpenGL foi produzida com C e C++, mas pode ser utilizada para diversas outras linguagens com um alto nível de eficiência.

O nosso simulador inicialmente estava parametrizado para o fluido água (veja Figura 4).

4.9 Condição de fronteira e Teste de colisão

Originalmente, o SPH foi concebido para tratar de problemas no domínio astrofísico, como o choque entre estrelas no espaço. Problemas dessa natureza tem condições de fronteira menos restritivas, quase inexistentes. Não havia a necessidade de estabelecer certas condições de contorno (VASCO, 2014).

² sistema multiplataforma que gerencia e automatiza o processo de “*build*”. Além disso, é um software gratuito e open-source.

³ <https://www.opengl.org/>

Figura 4 – Simulador SPH em execução com o fluido água, em 3 momentos: na figura (a), observamos o fluido discretizado em partículas. No momento (b), aplicamos uma das técnicas de visualização de superfícies (seção 4.11); e finalmente em (c) observamos a água que “bate” na taça vermelha (modelada por uma malha triangular) e é “derramada”, testando o algoritmo de colisão com fronteiras (seção 4.9)

Fonte: adaptado de Bísaro (2018)

Entretanto, na grande maioria dos problemas em Engenharia, lida-se com condições de contorno mais restritivas (impermeabilidade de paredes, condição de aderência, etc). Tais condições de contorno não são satisfeitas automaticamente pelas equações do método. Dessa forma, o tratamento das fronteiras recebe atenção especial e segue basicamente duas vertentes: a adoção das partículas fantasma (*ghost particles*) ou fronteiras reativas (*boundary forces*).

No caso das partículas fantasma, quando uma partícula aproxima-se da fronteira, há o espelhamento da partícula, criando uma partícula fictícia com a mesma distância normal à parede. Essa partícula fictícia, ou partícula fantasma, possui as mesmas características da partícula real, entretanto o sentido da velocidade segue a condição de contorno da parede. Esse artifício aumenta a vizinhança das partículas próximas à parede, garantindo unicidade do somatório do núcleo de suavização. A figura 5 exemplifica como são criadas as partículas fantasma para simular uma condição de contorno.

Figura 5 – Princípio das partículas fantasmas

Fonte: adaptado de Vasco (2014)

De acordo com a condição de impenetrabilidade, a componente normal da velocidade da partícula fantasma deve ser contrária à da partícula real. Sendo assim:

$$\vec{u}_{ghost} \cdot \vec{n} = -\vec{u}_i \cdot \vec{n} \quad (51)$$

Monaghan (1994), por sua vez, propôs uma técnica de controle de colisão das partículas com a fronteira rígida representando a fronteira através de partículas virtuais denominadas fantasmas. A ideia do uso dessas partículas (ilustradas em vermelho na

Figura 6) era gerar forças altamente repulsivas de maneira a prevenir a interpenetração do fluido nas fronteiras sólidas.

Figura 6 – Fronteira rígida com partículas fantasmas

Fonte: adaptado de Neto (2007)

A força de repulsão Γ_{ig} que atua sobre uma partícula i que colide com uma partícula fantasma g utilizada por Monaghan (1994) é uma adaptação do potencial de Lennard-Jones da dinâmica molecular, dada pela expressão:

$$\Gamma_{ig} = \begin{cases} D \left[\left(\frac{r_0}{r_{ig}} \right)^{12} - \left(\frac{r_0}{r_{ig}} \right)^4 \right] \frac{X_{ig}}{r_{ig}^2}, & \text{se } \left(\frac{r_0}{r_{ig}} \right) \leq 1 \\ 0, & \text{se } \left(\frac{r_0}{r_{ig}} \right) > 0 \end{cases} \quad (52)$$

onde $X_{ig} = X_i - X_g$, $r_{ig} = \|X_i - X_g\|$ e X_i se refere à partícula i , enquanto X_g à partícula correspondente fantasma (ghost). A constante D deve ser estimada na mesma ordem de grandeza do quadrado da norma da maior velocidade esperada no escoamento. De acordo com Neto (2007), o valor de r_{ig} desempenha um importante papel na composição da força de repulsão e deve ser escolhido como, aproximadamente, o valor da distância inicial entre as partículas.

No entanto, modelar geometrias complexas utilizando essa técnica de partículas fantasmas não se configura tarefa trivial, além do que o número de partículas a serem processadas aumenta consideravelmente, impactando no desenvolvimento computacional.

Por isso, usamos num primeiro momento uma abordagem estritamente geométrica (fronteira reativa), na qual a fronteira da superfície é representada por uma malha triangular

devidamente orientada. Implantamos um teste de colisão entre triângulos e partículas, adaptado de Neto (2007).

Este teste possui duas etapas distintas: a etapa de detecção de colisão, e a de atualização do estado da partícula. A Figura 7 ilustra um exemplo de resposta coerente à colisão de uma partícula com um triângulo, em função do vetor normal do mesmo.

Figura 7 – Colisão de uma esfera (partícula) com um triângulo (fronteira)

Fonte: adaptado de Neto (2007)

Para cada partícula i e cada triângulo T , se o produto escalar $v_i \cdot N_T \geq 0$, T é descartado, pois esse resultado significa que a partícula i não está em rota de colisão com T . Caso contrário, precisamos computar o tempo de colisão, que denotamos por t_c , da seguinte maneira:

$$t_c = \frac{\|(X_i - V_T) \cdot N_T\| - r_i}{\|v_i \cdot N_T\|} \quad (53)$$

onde V_T é um dos vértices do triângulo T e r_i é o raio da partícula i . Somente iremos para o próximo passo do procedimento se $t_c \leq \Delta t$. Neste caso, posicionamos a partícula i no seu provável ponto de colisão $X'_i = X_i + t_c v_i$ (método de Euler). Neste ponto, é realizado o teste de colisão propriamente dito, que é feito em três etapas (ver Figura 8).

Na primeira etapa da verificação de colisão da esfera com o triângulo, testamos se algum dos vértices de T está no interior da esfera de centro X'_i e raio r_i , caso negativo passamos para o segundo teste, que consiste em projetar o centro da esfera no plano definido por T , a partir da equação $X_{proj} = X'_i - [(X'_i - V_T) \cdot N_T] N_T$ e verificar se o X_{proj}

Figura 8 – Teste de colisão; a) tempo de colisão t_c ; b) colisão com um dos vértices de T ; c) colisão com o interior de T ; d) colisão com uma das arestas de T .

Fonte: adaptado de Neto (2007)

encontra-se no interior do triângulo T . Caso negativo, o último dos testes (colisão da esfera com uma aresta do triângulo) é realizado. Em caso negativo, não houve intersecção.

Considere a aresta $V_T W_T$ do triângulo T , com o objetivo de decidir se esta aresta intersecta a esfera e_i de centro X_i e raio r_i . Determinamos se o ponto r , que é a intersecção da reta que passa por X_i e é perpendicular à $V_T W_T$, encontra-se dentro da referida esfera. O ponto r (Figura 9) pode ser determinado pela equação

$$r = V_T + ud, \quad \text{com } d = W_T - V_T \text{ e } u = \frac{d \cdot X_i - d \cdot V_T}{\|d\|} \quad (54)$$

No algoritmo 2, explicamos em linhas gerais como funciona esse método de colisão de partículas.

Figura 9 – Ponto r , que determina se a esfera e_i intersecta a aresta $V_t W_t$

Fonte: adaptado de Neto (2007)

Se considerarmos $d^2 = \|r - X_i\|$, basta verificar se $d^2 \leq r_i^2$ e se $u \in [0, 1]$. O teste de colisão completo está apresentado no Algoritmo 2.

Algoritmo 2 Algoritmo básico de colisão

```

1: para cada partícula  $i \leftarrow 1$  até  $N$  faça
2:   para cada Triângulo  $T \leftarrow 1$  até  $M$  faça
3:     se  $v_i \cdot N < 0$  então
4:       Calcule  $X'_i = X_i + t_c v_i$ 
5:       se Existe intersecção entre  $X'_i$  e  $T$  então
6:         Calcule a velocidade da resposta  $v'_i$ ;
7:         Atualize  $X_i = X'_i + (\Delta t - t_c) v'_i$ ;
8:       fim se
9:     fim se
10:   fim para
11: fim para

```

Fonte: Neto (2007)

Vale ressaltar que a velocidade de resposta v'_i (linha 6 do algoritmo 2) é calculada em função do vetor normal do triângulo T (denotado como N_T). A colisão pode ser ou não perfeitamente elástica, sendo computada como:

$$v'_i = v_i + 2\lambda(v_i \cdot N_T) \quad (55)$$

onde $\lambda \in [0, 1]$ representa o coeficiente de restituição. Se $\lambda = 1$, a colisão é perfeitamente elástica.

No capítulo 6 retomaremos a discussão sobre Colisão com Fronteiras. Discutiremos alguns problemas detectados com o atual teste, e apresentaremos nossa proposta de teste para resolver estes problemas.

4.10 Incompressibilidade

Um fluido é dito *incompressível* quando a densidade de um elemento de fluido não é afetada pela variação da pressão, ou seja, durante o movimento o volume de qualquer elemento de fluido é preservado mesmo sobre influência da pressão. Pode-se aumentar ou diminuir a pressão, que a densidade permanecerá constante.

Assim, em um escoamento de fluido incompressível, a variação temporal da densidade ρ é **nula**, isto é,

$$\frac{D\rho}{Dt} = 0$$

Dessa forma, a equação 1, da continuidade, toma a simples forma:

$$\nabla \cdot \mathbf{v} = 0 \quad (56)$$

A restrição 56 é dita condição de incompressibilidade.

Matematicamente, um escoamento é incompressível quando o campo de velocidade possui divergência nula. A densidade de um elemento de fluido pode variar em virtude da condução térmica; porém as variações da temperatura são consideradas "pequenas" o suficiente para que as alterações na densidade sejam consideradas desprezíveis.

Uma classe relevante de fluidos incompressíveis é a classe dos líquidos (onde também se insere o sangue). Entretanto, no mundo real nenhum fluido é realmente incompressível, pois até mesmo um líquido pode ter a sua densidade aumentada aplicando-se uma pressão suficiente (PAIVA et al., 2009).

Uma outra forma de classificar um escoamento é através da velocidade do som no fluido. Fisicamente, a velocidade do som é a velocidade com a qual uma perturbação de pressão de pequena amplitude se desloca através de um fluido.

A razão entre a magnitude da velocidade do escoamento e a velocidade do som no fluido é conhecida como *número de Mach*, e é usualmente denotado como Ma . Para

números de Mach pequenos, $Ma < 0,3$; o escoamento é incompressível, caso contrário é dito compressível.

O método SPH foi inicialmente desenvolvido para simular fluidos compressíveis. A pressão é uma função explícita da densidade local do fluido. Apesar disso, podemos aproximar um fluido incompressível (como o sangue) para um fluido *quase-incompressível*.

Como vimos na Seção 4.2 do Capítulo 4, Monaghan (1994) usou a equação de estado (equação 34) para a modelagem de escoamentos com superfícies livres:

$$p_i = B \left(\left(\frac{\rho_i}{\rho_0} \right)^\lambda - 1 \right)$$

Onde ρ_0 é a densidade do fluido em repouso, B e λ são parâmetros relacionados às flutuações do fluido.

O parâmetro B é o termo relacionado às flutuações de densidade do fluido, e é estimado por:

$$B = \frac{c^2 \rho_0}{\lambda}$$

Onde c é a velocidade do som. A velocidade do som representa a velocidade pontual mais rápida de propagação de onda naquele meio - em SPH, ela é escolhida como sendo aproximadamente dez vezes a maior velocidade esperada no escoamento do fluido, isto é, um número de Mach em torno de 0,1 para manter a condição de incompressibilidade.

4.10.1 WCSPH

Para conseguir simular um fluido incompressível, como o sangue; aproximamos-o de um fluido *quase-incompressível*: essa técnica é chamada de *WCSPH* - *Weakly or quasi-Compressibility SPH*, e foi desenvolvida por Monaghan (1994).

Existe uma outra abordagem de se obter essa incompressibilidade com o SPH, chamado de *ISPH* - *truly Incompressible SPH* (KULASEGARAM; KARIHALOO; GHANBARI, 2011).

O método WCSPH é geralmente mais utilizado por sua versatilidade de usos, e por sua formulação explícita, que favorece uma implementação mais simples (o método ISPH é implícito) (AL-SAAD, 2017). Além disso, apresenta uma boa acurácia de resultados para vários fluidos e fronteiras com geometrias complexas.

De acordo com Solenthaler e Pajarola (2009), o uso da equação de Tait pode levar ao aparecimento de oscilações numéricas no campo de pressões, o que introduz uma instabilidade no método. Além disso, a necessidade da escolha de uma alta velocidade do som, para efeito de resolução numérica, pode levar a severas restrições no passo de tempo em razão da condição de Courant-Friedrichs-Lewy (CFD).

Para resolver esse problema, e de acordo com a equação da continuidade (Equação 1), a atualização da densidade é feita iterativamente da seguinte forma:

$$\rho_i^{n+1} = \rho_i^n - \rho_i^n (\nabla \cdot \mathbf{v}_i^{n+1}) \Delta t^{n+1} \quad (57)$$

Com isso, a pressão de todas as partículas no passo de tempo $n + 1$ pode ser calculada usando-se a equação de estado de Tait (Equação 34).

4.11 Visualização de superfícies

O problema da visualização realística de simulações baseadas em partículas pode ser estabelecido como o de gerar uma superfície linear por partes, na maioria das vezes representadas por triângulos, que se “encaixe” no bordo da nuvem de pontos que representa o volume de fluido. A dificuldade que se apresenta na geração desta superfície é que, devido a ausência de conectividade entre as partículas, torna-se difícil identificar as partículas que estão no bordo do volume (BÍSCARO, 2018).

Existem duas principais classes de técnicas para a visualização de superfícies baseadas em simulações físicas: técnicas advindas de zeros de funções (também conhecidas como técnicas de *Level Set*), e técnicas baseadas em projeções no chamado “sistema de coordenadas de câmera”.

Nosso simulador está apto para trabalhar com os dois tipos de técnicas. Ele suporta ao todo 6 algoritmos diferentes de visualização de superfícies (5 da classe *Level Set*), e 1 da classe de projeção no sistema de coordenadas de câmera), dos seguintes autores:

- Müller, Charypar e Gross (2003)
- Zhu e Bridson (2005)
- Solenthaler, Schläfli e Pajarola (2007)
- Onderik, Chládek e Durikovic (2013)
- Yu e Turk (2013)

- Sandim et al. (2016)

Na figura 10 apresentamos um comparativo entre os algoritmos de visualização baseadas em zeros de função. Para isso, congelamos um instante comum da simulação, e fizemos a geração de superfície com cada um dos 5 algoritmos.

4.12 Parâmetros da Simulação

Durante a simulação, foram pré-definidos alguns parâmetros para que obtivéssemos realismo na simulação de um vaso sanguíneo real. Os parâmetros utilizados neste trabalho são apresentados na Tabela 1.

Tabela 1 – Parâmetros usados em nossa implementação

Parâmetro	Valor	Descrição
h	0,084	Raio de influência de cada partícula
ρ_0	1060	Densidade (padrão) do Sangue
k	1400	Compressibilidade
γ	1	Pressão
ν	0,025	Viscosidade
g	(0;-9,8;0)	Aceleração da Gravidade
q	57	Quantidade de Vizinhos por Partícula

Fonte: adaptado de Al-Saad (2017)

4.13 Simulação de uso

Nesta seção, apresentamos algumas imagens do nosso simulador em funcionamento, com o fluido sangue.

A malha triangular que representa o nosso vaso sanguíneo tem ao todo 194 vértices, e 394 triângulos (veja Figura 12). As malhas que usamos estão no formato OFF (*Object File Format*), um tipo de arquivo que armazena objetos geométricos em 3D.

Na Figura 11 temos uma demonstração completa do simulador, operando com sangue. As imagens foram obtidas em 4 instantes de tempos. Foram usadas 7.967 partículas, e o tempo total da simulação foi de 120 s.

Para geração destas imagens, foi usado um algoritmo de traçado de raios (Ray Tracing), uma técnica usada para síntese (renderização) de imagens 3D. O algoritmo Ray Tracing inverte o sentido do trajeto dos raios de luz, como se o observador fosse a fonte de

Figura 10 – Comparação de diferentes técnicas de visualização para o mesmo instante de uma simulação. Em (a), temos o fluido discretizado em partículas. Em (b), a superfície foi gerada considerando a técnica de Müller, Charypar e Gross (2003), em (c) por Zhu e Bridson (2005), em (d) por Onderik, Chládek e Durikovic (2013), em (e) por Yu e Turk (2013) e em (f) por Sandim et al. (2016).

Figura 11 – Simulador SPH em funcionamento com o fluido sangue, em 4 momentos: na figura (a) temos o início da simulação, que passa pelos instantes intermediários (b) e (c) e finaliza em (d). As imagens finais foram geradas com um algoritmo de traçado de raios.

(a)

(b)

(c)

(d)

Fonte: Thiago Sousa, 2019.

Figura 12 – Arquivo .OFF do vaso sanguíneo com uma bifurcação, modelado por malha triangular. A visualização foi obtida pelo software MeshLab.

Fonte: Thiago Sousa, 2019.

luz - desta forma, são processados apenas os raios efetivamente vistos pelo observador. O resultado é uma imagem com muito mais realismo.

A simulação exibida nesta sessão foi realizada num computador com processador Intel(R) Core(TM) i7-4500U com quatro núcleos 1.80GHz e com 8 GB de memória RAM, sistema operacional Ubuntu 18.04.

5 Validação do simulador

Para validar nosso simulador, e verificar sua acurácia e precisão na simulação do sangue, usamos como *benchmark* o teste mais mencionado na literatura científica desta natureza: o teste de fluxo de Poiseuille.

Jean Louis Marie Poiseuille (1797-1869) foi um médico e físico francês que formulou uma lei (equação 58) para o escoamento de um fluido viscoso através de um canal cilíndrico (como o sangue que corre em um vaso arterial). Segundo essa Lei, a velocidade do escoamento é diretamente proporcional à pressão do fluido e ao raio do tubo, e inversamente proporcional ao comprimento do tubo e à viscosidade do fluido:

$$Q = \frac{P\pi R^4}{8l\eta} \quad (58)$$

onde Q é a vazão do tubo cilíndrico, de raio R e comprimento l , que transporta um líquido de pressão P e com coeficiente de viscosidade η (RAPP, 2017).

O teste de fluxo de Poiseuille tem a seguinte estrutura: um tubo de comprimento infinito é representado por duas placas infinitas, uma na posição $y = 0$ e a outra na posição $y = L$ (L é o diâmetro do tubo). O movimento do fluido acontece devido a diferença de pressão na direção horizontal, representado em nosso simulador pela força f_b . Para a realização deste teste, qualquer outra força na direção vertical foi desconsiderada.

Com o passar do tempo, o resultado esperado é que as partículas que estão no centro do tubo alcancem a maior velocidade na direção horizontal (eixo X), e que a velocidade decresça simetricamente conforme se aproxima das fronteiras (as paredes do tubo), chegando a zero nestes casos.

Um fluido incompressível que satisfaz essas condições é chamado de fluido laminar, pois cada camada de partículas se comporta como se fosse uma lâmina diferente (Figura 13). Fluidos laminares tem um baixo (menor do que 2000) número de Reynolds, grandeza definida a seguir na seção 5.1. (AL-SAAD, 2017)

Figura 13 – Representação gráfica do Teste de Fluxo de Poiseuille

Fonte: adaptado de Vukcevic, Werner e Degiuli (2012)

5.1 Número de Reynolds

O número de Reynolds (Re) é uma grandeza adimensional, usado em mecânica dos fluidos para o cálculo do regime de escoamento de determinado fluidos da seguinte forma (Equação 59):

$$Re = \frac{VL}{\nu} \quad (59)$$

onde V é a máxima velocidade estável do fluido, L é o diâmetro do tubo e ν é viscosidade cinemática do fluido.

O número de Reynolds é usado para avaliar a estabilidade de um fluxo, indicando se o fluido flui de forma laminar ou turbulenta.

De forma geral, para números de Reynolds menores que 2.000 o fluxo é considerado laminar, e para valores maiores que 2.400 o fluxo é considerado turbulento. Valores entre 2.000 e 2.400 têm um fluxo transitório (MORRIS; FOX; ZHU, 1997).

Num escoamento laminar, as partículas do fluido se movimentam ao longo de trajetórias bem definidas, apresentando lâminas ou camadas que preservam suas características durante o escoamento (Figura 14). Nesse tipo de escoamento, a viscosidade age no fluido no sentido de amortecer a tendência de surgimento da turbulência.

Já o escoamento turbulento ocorre quando as partículas de um fluido não se movimentam ao longo de trajetórias bem definidas, ou seja, as partículas descrevem trajetórias irregulares, com movimento aleatório (Figura 15), produzindo uma transferência de quantidade de movimento entre regiões de massa líquida.

Figura 14 – Exemplo de fluido laminar

Fonte: adaptado de Kundu, Cohen e Dowling (2016)

Figura 15 – Exemplo de fluxo turbulento

Fonte: adaptado de Kundu, Cohen e Dowling (2016)

A solução analítica para o cálculo da velocidade axial (na direção do eixo X) de um fluido laminar incompressível, em função do tempo t e da altura y da lâmina da partícula é dada pela Equação 60 (MORRIS; FOX; ZHU, 1997):

$$V_x(y, t) = \frac{f_b}{2\nu} y(y - L) + \sum_{n=0}^{\infty} \frac{4f_b L^2}{\nu \pi^3 (2n+1)^3} \sin\left(\frac{\pi y}{L} (2n+1)\right) \exp\left(-\frac{(2n+1)^2 \pi^2 \nu}{L^2} t\right) \quad (60)$$

onde V_x é a velocidade do fluido na direção x , t é o tempo e ν a viscosidade específica.

A força f_b é calculada pela equação 61:

$$f_b = \frac{8\nu V}{L^2} \quad (61)$$

onde V é a máxima velocidade estável.

5.2 Descrição do Teste

Para o nosso teste, usamos uma versão simplificada em 2D do nosso simulador, como sugere a metodologia de Al-Saad (2017). Nós removemos todas as forças que interagem na

direção vertical (eixo Y). O fluido, então com uma quantidade infinita de partículas, escoar em um tubo que também é considerado de comprimento infinito.

Na posição $x = 0,005m$ nós fixamos nosso ponto de observação: a cada intervalo de tempo, nós tomamos a velocidade das partículas que passam por este ponto, para cada altura y das lâminas do fluido.

Finalmente, nós comparamos as velocidades V_x encontradas com o simulador com aquelas calculadas analiticamente pela equação 60.

Para verificar também a interferência da força f_b na velocidade de deslocamento das partículas (equação 61), nós realizamos o experimento em dois cenários diferentes: assumindo uma velocidade máxima estável (V) como $0,01m/s$; e assumindo uma velocidade 10 vezes maior ($V = 0,1m/s$).

Estes foram os parâmetros comuns utilizados em nossas simulações:

- $L = 0.001m$
- $\rho = 1000kg/cm^3$
- $\nu = 1 \times 10^{-6}m^2s^{-1}$
- 861 partículas
- $\Delta x = \Delta y = 0.0020m$

5.3 Resultados e discussões

O primeiro teste que fizemos foi para $V = 0,01m/s$. O diâmetro L foi particionado em 51 níveis de altura. Tomamos as velocidades (em m/s) em 3 momentos diferentes: $t = 0,05s$, $t = 0,20s$ e $t = 0,50s$.

Na Figura 16 plotamos, em um mesmo sistema cartesiano, o gráfico das velocidades calculadas pela solução analítica (linhas contínuas), segundo a equação 60; e o gráfico das velocidades observadas com o nosso simulador (linhas com marcadores).

Depois, realizamos o mesmo teste para o caso em que $V = 0,1m/s$, e obtivemos as velocidades nos passos de tempo $t = 0,05s$, $t = 0,15s$ e $t = 0,25s$. As alturas são idênticas ao experimento anterior, totalizando 51 níveis.

Na Figura 17 plotamos o gráfico de dispersão das velocidades observadas (linhas com marcadores) e calculadas (linhas contínuas).

Figura 16 – Comparação entre os resultados observados e os calculados, considerando $V = 0,01m/s$.

Fonte: Thiago Sousa, 2019.

Para comparar a efetividade dos resultados observados com aqueles obtidos através da solução analítica, calculamos o Erro Relativo segundo a fórmula 62:

$$Erro\_Relativo = \sqrt{\frac{\sum_{i=1}^N (v_{i,SPH} - v_{i,Exato})^2}{\sum_{i=1}^N (v_{i,Exato})^2}} \quad (62)$$

onde $v_{i,SPH}$ e $v_{i,Exato}$ são, respectivamente, a velocidade observada pelo simulador SPH e a calculada pela solução analítica para a partícula 'i', e N é o número total de partículas usadas no cálculo do Erro.

Para o caso em que $V = 0,01m/s$, o erro foi de apenas 1,57%. No teste em que $V = 0,1m/s$, o erro relativo foi de 4,80%. Um erro menor que 5% garante a validade do nosso simulador, que está em consonância com os resultados esperados encontrados na literatura (AL-SAAD, 2017).

Figura 17 – Comparação entre os resultados observados e os calculados, considerando $V = 0,1 \text{ m/s}$.

Fonte: Thiago Sousa, 2019.

Podemos observar também que, aumentando a força aplicada (que é diretamente proporcional à velocidade máxima estável V), os dados experimentais convergem ainda mais para os resultados teóricos.

6 Teste de Colisão com Rotações

Na seção 4.9, vimos o funcionamento do teste de colisão com fronteira que usamos inicialmente, baseado na tese de Neto (2007).

Daquele método, identificamos alguns problemas que comprometem a credibilidade da simulação:

- O cálculo da posição final da partícula, após a colisão, é discreto; mas o passo de tempo Δt não é tão pequeno (note que usar um passo de tempo infinitesimalmente menor poderia tornar a simulação impraticável, pois seriam necessários muitos ciclos para o cálculo de poucos segundos). Com isso, na passagem entre um passo de tempo e outro, uma partícula pode ter “escapado” da malha, e ela não retornará mais ao seu interior, tendo ocorrido então um vazamento (veja Figura 18).

Figura 18 – A partícula i em rota de colisão com a aresta de um triângulo da malha (a). A velocidade e a posição da partícula é calculada (b). A partícula tem o seu curso alterado, mas a nova posição escapa do triângulo de baixo (c). A partícula i “vazou” da malha.

Fonte: Thiago Sousa, 2019.

- Também existem casos em que o vetor velocidade da partícula é paralelo à face do triângulo; o que faz com que o ângulo entre o vetor velocidade da partícula e o vetor normal do triângulo da malha se aproxime de 90° . Isso causa um erro numérico no produto interno (linha 3 do Algoritmo 4.9), que tenderá a zero, criando falsos-positivo ou falsos-negativos na detecção de colisão com a borda.

Na figura 19 reproduzimos a clássica simulação do fluido que escorre em uma caixa cúbica (dam break), caixa esta também modelada com uma malha triangular (SHAO; LO, 2003). Neste caso, é fácil observar os efeitos das imperfeições do Algoritmo 4.9. Algumas partículas acabam escapando, especialmente as que estão perto das arestas dos triângulos.

Importante destacar, também, que o “vazamento” não ocorre com todo o fluido. Para fluidos de alta viscosidade, inclusive, esse efeito nem será observado.

Figura 19 – Simulação Dam Break. Esta simulação tinha ao todo 15.348 partículas, mas 39 delas “vazaram” da malha pelas arestas dos triângulos.

Fonte: Thiago Sousa, 2019.

Dado essa situação, foi necessário aprimorar o teste de colisão com fronteiras para termos um algoritmo que solucionasse esses casos específicos; criando assim uma simulação mais confiável e sem o vazamento de partículas.

6.1 Descrição do novo teste de colisão

Considere um triângulo T_j da malha, definido pelos seus vértices P_1, P_2 e P_3 e uma partícula i qualquer do fluido (Figura 20).

Figura 20 – Uma partícula i do fluido, com velocidade v_i , em rota de colisão com o triângulo T_j da malha.

Fonte: Thiago Sousa, 2019.

Uma quantidade considerável dos vazamentos que observamos ocorrem após as colisões, quando as partículas “pulam” para fora da malha. Para corrigir este problema, nós aplicamos uma série de transformações geométricas nos elementos do teste (triângulo e partícula) que, além de simplificar o teste, ainda corrige o problema de vazamento.

A ideia deste novo teste é transportar o conjunto triângulo / partícula (veja Figura 21) que será testado para a origem do sistema de coordenadas no $\mathbb{R}^3$; e então aplicar uma sequência de rotações para alinhar o vetor normal do triângulo com o eixo Y , e uma de suas arestas com o eixo Z (veja Figura 22).

Figura 21 – Transportar os vetores para a Origem.

Fonte: Thiago Sousa, 2019.

Figura 22 – Aplicar a sequência de rotações ao par (triângulo T_j + partícula i).

Fonte: Thiago Sousa, 2019.

Essa sequência de transformações, conforme vemos na linha 7 do Algoritmo 3, consiste em primeiro rotacionar o conjunto para o plano YZ, e depois rotacioná-lo novamente para o plano XZ.

Para realizar essas rotações em 3D, nós usamos os números quatérnios. A construção dos quatérnios já favorece este tipo de aplicação; como vimos na Seção 3.4.

6.2 Teste de Colisão Rotacionada

Após algumas rotações, o novo Algoritmo que desenvolvemos é capaz de avaliar se houve uma colisão - e, caso haja, ele pode resolvê-la.

Os principais passos de nosso teste estão no Algoritmo 3:

Algoritmo 3 Teste de Colisão Rotacionada

```

1: para cada partículas  $i \leftarrow 1$  até  $N$  faça
2:   para cada Triângulos  $T_j, j \leftarrow 1$  até  $M$  faça
3:     se  $v_i.N < 0$  então
4:       Calcule a distância  $d$  da partícula  $i$  até  $T_j$ ;
5:       se distância  $< (\delta * r)$  então
6:          $r$  é o raio da partícula;
7:         Aplicar as transformações geométricas, com os quatérnios;
8:         se TesteColisãoRotacionada ( $T_j, i$ ) == VERDADEIRO então
9:           Calcule a nova trajetória;
10:          Aplique as transformações reversas na partícula  $i$ ;
11:        fim se
12:      fim se
13:    fim se
14:  fim para
15: fim para

```

Fonte: Thiago Sousa, 2019.

O produto escalar executado na linha 3 do Algoritmo 3 tem a função de eliminar as partículas que não estão em rota de colisão com o plano/face do triângulo T_j . A distância d calculada na linha 4 selecionará somente as partículas que estejam perto de T_j . O valor de δ está parametrizado entre 3 e 5 (linha 5 do algoritmo).

Depois dessas verificações, uma sequência de transformações geométricas é aplicada ao par triângulo / partícula, com o objetivo de posicionar o triângulo no plano XZ e tendo uma de suas arestas alinhadas ao eixo Z, como exibido na Figura 22.

Deste modo, o teste de colisão é reduzido para uma simples comparação de coordenadas (se a coordenada Y da partícula é negativa), seguido de uma avaliação de duas

equações de retas (a reta definida pelos vértices P_1 e P_2 do triângulo; e a reta definida pelos vértices P_3 e P_2), que agora estão no plano XZ (veja Figura 21).

Se o teste de colisão foi positivo, a partícula i está do “lado de fora” do triângulo T_j , então a nova posição da partícula é calculada e seu vetor velocidade atualizado. Após isso, aplicamos as inversas das transformações no vetor velocidade, para que a partícula prossiga com seu movimento no interior da malha.

Observamos também que, mesmo que algumas partículas “escapem” da malha, no próximo passo de tempo elas serão detectadas na comparação com algum triângulo vizinho à T_j .

6.3 Resultados

Para verificar a eficiência do novo algoritmo de colisão, nós criamos 3 casos de teste:

- teste de colisão dentro de uma caixa cúbica (dan break). Nesse teste, a caixa (sem tampa) é modelada por uma malha triangular composta de 10 triângulos retângulos, e 8 vértices. A Figura 23 mostra a execução do teste com o Algoritmo de Colisão originalmente implementado (baseado no trabalho de Neto (2007)); e o mesmo teste com a execução do Algoritmo de Colisão Rotacionada.
- colisão em um vaso sanguíneo com 1 bifurcação. Aqui, a malha triangular que modela este vaso é composta de 194 vértices e 339 triângulos. A Figura 24 mostra a execução do teste com o Algoritmo de Colisão originalmente implementado (baseado no trabalho de Neto (2007)); e o mesmo teste com a execução do Algoritmo de Colisão Rotacionada.
- colisão em um vaso sanguíneo com 3 bifurcações. Este vaso foi modelado por uma malha triangular de 236 vértices e 409 triângulos. A Figura 25 mostra a execução do teste com o Algoritmo de Colisão originalmente implementado (baseado no trabalho de Neto (2007)); e o mesmo teste com a execução do Algoritmo de Colisão Rotacionada.

Para cada um dos casos de teste, o experimento consistia na simulação com o tradicional algoritmo de colisão, e com o novo algoritmo que desenvolvemos; seguido da contagem de partículas que escaparam em cada um dos casos.

Figura 23 – Teste em uma caixa cúbica, usando o Algoritmo Tradicional (a) e o Algoritmo de Colisão Rotacionada (b). As cores das partículas indicam a intensidade de suas velocidades.

Figura 24 – Teste no vaso sanguíneo com 1 bifurcação. Em (a), foi usado o Algoritmo Tradicional. Note a quantidade de partículas que “escapam”, especialmente na bifurcação do meio. Em (b) foi usado o Algoritmo de Colisão Rotacionada. Observe que não são encontrados “vazamentos”.

Figura 25 – Teste em um vaso com 3 bifurcações. Em (a), usando o Algoritmo Tradicional, pode ser visto que algumas partículas acabam “escapando” da malha triangular. Em (b), foi usado o Algoritmo de Colisão Rotacionada. Em ambos os casos, as cores das partículas indicam a intensidade de suas velocidades.

(a)

(b)

Para checar os vazamentos, nós criamos uma função que conta o número de partículas que escaparam ao final da simulação.

Na tabela 2 apresentamos os resultados para esses 3 casos de teste.

Tabela 2 – Resultados do Teste para vazamento de partículas, comparando o algoritmo originalmente utilizado (baseado na obra de Neto (2007)), e o Algoritmo de Colisão Rotacionada que implementamos.

Caso de Teste	Qtd Part.	Alg. Tradic.	Alg. Rotação
Dan break	15340	58	0
Vaso com 1 bifurcação	5154	65	0
Vaso com 3 bifurcações	4812	79	0

Fonte: Thiago Sousa, 2019.

Em nossos testes, observamos que o algoritmo de colisão rotacionada foi eficiente para conter as partículas que escapavam.

Outra característica que observamos é que conforme aumentamos o número de bifurcações em um vaso sanguíneo, a tendência é que também aumente a quantidade de partículas que vazam.

6.4 Discussões Finais

A colisão com fronteira é um fator importante no desenvolvimento de uma simulação, já que interfere diretamente na confiabilidade e realismo da simulação. É essencial que num sistema que simula a circulação do sangue em uma artéria não ocorram “vazamentos”.

O algoritmo de colisão rotacionada que implementamos simplificou o teste de colisão, pois reduziu o problema que antes era 3D, para uma situação em 2D.

O método se mostrou eficiente para corrigir o vazamento de partículas. Graças a implementação das rotações com os números quatérnios, o algoritmo foi capaz de manter a estabilidade do simulador sem comprometer sua performance (a implementação com o novo teste de colisão teve um tempo de execução praticamente igual ao da implementação anterior).

Essa nova metodologia para o tratamento de colisão de partículas com fronteiras, totalmente geométrico, pode ser utilizado em outros sistemas de simulação baseado em partículas (lagrangianos), sem a necessidade de se adotar partículas fantasmas ou introduzir novos elementos à simulação.

Em particular, para o caso de aplicações médicas - como em nosso simulador do sangue, este novo teste torna a simulação mais realista e confiável, algo que é imprescindível para sistemas dessa natureza.

7 Conclusões

Simular computacionalmente o sangue é uma atividade importante. Esta atividade tem uma ampla gama de uso, e pode servir para o treinamento médico de cirurgias minimamente invasivas, sistemas de apoio à decisão para médicos, desenvolvimento de novos equipamentos médicos, investigação de doenças, entre outras aplicações.

Dentre as diversas formas para se fazer esta simulação, o método lagrangiano SPH se destaca pela liberdade de não depender de malhas, e a adaptabilidade à diversos tipos de geometria - embora a contrapartida seja um processamento computacional robusto; pois a cada passo de tempo muitos cálculos precisam ser feitos.

Além de definir os conceitos de mecânica de fluidos e apresentar os fundamentos do método SPH, usamos a técnica de Revisão Sistemática para encontrar o estado da arte no uso do método SPH para a simulação do sangue - o que apontou para a viabilidade de nossa pesquisa.

Adaptamos o simulador SPH para o uso com o fluido biológico sangue. Para tanto, discutimos neste trabalho sobre como formular de forma lagrangeana as equações de Navier-Stokes, as ressalvas sobre o cálculo da pressão, a forma em que é feita a integração temporal com a técnica leap-frog, a correção XSPH e o levantamento de parâmetros.

Para manter a condição de incompressibilidade do sangue, mostramos a atualização que deve ser feita na densidade de cada partícula e o método WCSPH; que na verdade o aproxima de um fluido quase-incompressível.

Usamos o teste mais referenciado na literatura - o de fluxo de Poiseuille para validar a acurácia numérica de nosso simulador, comparando resultados observados na prática com aqueles calculados segundo a fórmula analítica. O erro relativo total, menor que 5%, comprova que os resultados produzidos pelo simulador são confiáveis.

Uma contribuição relevante ao estado da arte também foi a proposição de um novo teste de colisão, rotacionado, construído com números quatérnios. Além de simplificar o teste de colisão com fronteiras sem comprometer sua performance, esse teste ainda conteve as partículas que “escapavam” da malha triangular. Acreditamos também que esse teste pode ser empregado em outros sistemas de simulação de fluidos baseado em partículas.

Nossa hipótese de pesquisa era a de que conseguiríamos usar o método SPH para adaptar um sistema de simulação computacional do sangue, e com um novo teste de colisão, contribuir com o estado da arte das simulações baseadas em partículas.

Os resultados de nosso trabalho confirmam a hipótese da nossa pesquisa.

A seguir, apresentamos algumas sugestões para aprofundamentos e trabalhos futuros.

7.1 *Trabalhos Futuros*

Neste trabalho, conseguimos atingir o objetivo proposto: adaptar um simulador SPH para o uso com um fluido biológico - o sangue.

Dada a versatilidade do método SPH, a sua estabilidade, e possibilidade de uso com malhas de geometrias complexas, elencamos algumas sugestões (porém não limitado a estes) de trabalhos futuros.

7.1.1 Paralelização do simulador

Uma das vantagens do método SPH é a de não precisar usar malhas para simular o fluido.

Por outro lado, a cada passo de tempo, e para cada partícula, uma grande quantidade de cálculos precisa ser feita pelo simulador.

Logo, para otimizar o processamento, pode ser estudado a aceleração dessas tarefas repetitivas, com o uso de Paralelização ou de outras técnicas.

Uma das técnicas de paralelização mais usadas nos dias atuais consiste no uso dos multi-processadores presentes nas placas gráficas, as chamadas GPUs (*Graphics Processing Units*). A fabricante NVidia tem popularizado a linguagem de programação CUDA (*Computer Unified Device Architecture*), própria para suas GPUs, que permite uma interface entre programação sequencial e a paralelizada.

Souto, Góes e Góes (2017) fizeram um experimento para a aceleração de um simulador SPH com essa tecnologia. Apesar dos bons resultados, o acesso às memórias e a transferência de dados entre CPU (*Central Processing Unit*) e GPU ainda é um problema relevante.

7.1.2 Visualização de superfícies

Um dos diferenciais de nosso simulador é a possibilidade de se usar diferentes técnicas para a visualização de superfícies, construídas a partir de um conjunto de partículas (Seção 4.11).

Mas qual deles produz o melhor efeito de visualização; qual o mais realista? Em que contexto, e para qual público?

Essa investigação também poderia ser um interessante tema de pesquisa, e que certamente contará com testes de avaliação por usuários.

7.1.3 Malhas elásticas

No nosso simulador, a malha triangular que reproduz os vasos sanguíneos é rígida.

Mas, no corpo humano, os vasos sanguíneos possuem alguma elasticidade.

Portanto, poderia ser verificado o comportamento do sangue em uma malha elástica, e a interação entre fluido e malha.

7.1.4 Simulação multi-fase

Graças a flexibilidade do simulador SPH em operar em sistemas com geometrias complexas, poderia se estudar a interação do sangue com outros fluidos.

Al-Saad, Kulasegaram e Bordas (2016), por exemplo, estudaram a simulação do sangue com os “trombos” que acabam formando a trombose, detalhando como ocorre essa formação e levantando alternativas (médicas) para combatê-las.

Assim também poderia ser investigado o surgimento e a relação com outras doenças do sistema circulatório, como infarto, aneurisma; e a discussão de exames como a cintilografia e o cateterismo.

7.1.5 Interação com dispositivos hápticos

Por último, como vimos na Seção 1.1, uma classe relevante de experimentos pode ser feita entre o simulador e instrumentos hápticos que possibilitam a interação com o sangue.

Por exemplo, a interação com um bisturi em ambiente de realidade aumentada, seringas, entre outros dispositivos médicos.

Referências¹

- AL-SAAD, M. *Blood flow simulation using Smoothed Particle Hydrodynamics*. Tese (Doutorado) — Cardiff University, 2017. Citado 5 vezes nas páginas 62, 64, 68, 70 e 72.
- AL-SAAD, M.; KULASEGARAM, S.; BORDAS, S. Blood flow simulation using smoothed particle hydrodynamics. In: *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*. [S.l.: s.n.], 2016. v. 4, p. 8241–8246. Citado 6 vezes nas páginas 22, 31, 32, 33, 34 e 87.
- ALVES, J. F. B. *Uma análise comparativa dos principais métodos baseados em partículas para simulação de escoamentos*. 96 p. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, 2008. Citado 2 vezes nas páginas 41 e 49.
- BIOLCHINI, J. et al. *Sytematic review in software engineering*. [S.l.], 2005. Citado 2 vezes nas páginas 24 e 26.
- BÍSCARO, H. H. et al. Comparing efficient data structures to represent geometric models for three-dimensional virtual medical training. *Journal of Biomedical Informatics*, v. 63, p. 195 – 211, 2016. ISSN 1532-0464. Disponível em: <http://www.sciencedirect.com/science/article/pii/S153204641630096X>. Citado na página 22.
- BÍSCARO, H. H. *Análise de Técnicas de Visualização de Superfícies para Simulações de Fenômenos Físicos Baseadas em Partículas*. 99 p. Tese (Livre Docência em Computação Gráfica) — Universidade de São Paulo. Escola de Artes, Ciências e Humanidades, 2018. Citado 5 vezes nas páginas 42, 54, 55, 63 e 65.
- COURANT, R.; FRIEDRICHS, K.; LEWY, H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, v. 100, p. 32–74, 1928. Citado na página 43.
- DAGENAIS, F.; GAGNON, J.; PAQUETTE, E. A prediction-correction approach for stable sph fluid simulation from liquid to rigid. In: *Proceedings of Computer Graphics International 2012*. [S.l.: s.n.], 2012. Citado na página 52.
- DESBRUN, M.; GASCUEL, M.-P. Smoothed particles: A new paradigm for animating highly deformable bodies. In: *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*. New York, NY, USA: Springer-Verlag New York, Inc., 1996. p. 61–76. ISBN 3-211-82885-0. Disponível em: <http://dl.acm.org/citation.cfm?id=274976.274981>. Citado na página 51.
- FRIGO, L. M. *Simulação numérica de escoamentos incompreensíveis tridimensionais turbulentos e em transição*. 139 p. Dissertação (Mestrado) — Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2004. Citado na página 21.
- GINGOLD, R. A.; MONAGHAN, J. J. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, v. 181, p. 375–389, nov. 1977. Citado 2 vezes nas páginas 37 e 40.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

GRØNLI, T. et al. Real-time assimilation and regularization of ultrasound blood velocity measurements using smoothed particle hydrodynamics. In: *2017 IEEE International Ultrasonics Symposium (IUS)*. [S.l.: s.n.], 2017. p. 1–1. Citado 3 vezes nas páginas 31, 33 e 34.

GUEDES, C. M. C. F. D. F. M. *Métodos sem Malha em Problemas de Mecânica Computacional. Aplicação a Processos de Enformação Plástica*. 205 p. Tese (Doutorado) — Universidade do Porto - Faculdade de Engenharia, 2006. Citado na página 37.

GUIDORIZZI, H. L. *Um curso de cálculo. Volume 3*. 5. ed. Rio de Janeiro: LTC, 2001. Citado na página 45.

HOLLAND, M. I.; POP, S. R.; JOHN, N. W. Vr cardiovascular blood simulation as decision support for the future cyber hospital. In: *2017 International Conference on Cyberworlds (CW)*. [S.l.: s.n.], 2017. p. 233–236. Citado 5 vezes nas páginas 22, 30, 31, 33 e 34.

HUANG, D. et al. Real-time simulation of contrast media diffusion based on gpu. In: *2015 International Conference on Virtual Reality and Visualization (ICVRV)*. [S.l.: s.n.], 2015. p. 286–289. Citado 3 vezes nas páginas 30, 31 e 32.

KITCHENHAM, B. *Procedures for performing systematic reviews*. [S.l.], 2004. Citado na página 26.

KULASEGARAM, S.; KARIHALOO, B.; GHANBARI, A. Modelling the flow of self-compacting concrete. *International Journal for Numerical and Analytical Methods in Geomechanics*, v. 35, p. 713 – 723, 04 2011. Citado na página 62.

KULP, S. et al. Practical patient-specific cardiac blood flow simulations using sph. In: *2013 IEEE 10th International Symposium on Biomedical Imaging*. [S.l.: s.n.], 2013. p. 832–835. ISSN 1945-7928. Citado 6 vezes nas páginas 29, 30, 31, 32, 33 e 34.

KUNDU, P. K.; COHEN, I. M.; DOWLING, D. R. *Fluid Mechanics*. Sixth edition. Boston: Academic Press, 2016. 409–467 p. ISBN 978-0-12-405935-1. Citado na página 70.

LIU, G.; LIU, M. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific, 2003. ISBN 9789812564405. Disponível em: https://books.google.com.br/books?id=_cwFMmEQvZQC. Citado 4 vezes nas páginas 38, 39, 41 e 49.

LUCY, L. B. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, v. 82, 1977. Disponível em: <http://adsabs.harvard.edu/full/1977AJ.....82.1013L>. Citado na página 37.

MILLER, G.; PEARCE, A. Globular dynamics: A connected particle system for animating viscous fluids. *Computer and Graphics*, v. 3, n. 13, p. 305–309, 1989. Citado na página 38.

MONAGHAN, J. On the problem of penetration in particle methods. *Journal of Computational Physics*, v. 82, n. 1, p. 1–15, 1989. ISSN 0021-9991. Disponível em: <http://www.sciencedirect.com/science/article/pii/0021999189900326>. Citado na página 52.

MONAGHAN, J. Simulating free surface flows with sph. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 110, n. 2, p. 399–406, fev. 1994. ISSN 0021-9991. Disponível em: <http://dx.doi.org/10.1006/jcph.1994.1034>. Citado 4 vezes nas páginas 48, 56, 57 e 62.

MORRIS, J. P.; FOX, P. J.; ZHU, Y. Modeling low reynolds number incompressible flows using sph. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 136, n. 1, p. 214–226, set. 1997. ISSN 0021-9991. Disponível em: <http://dx.doi.org/10.1006/jcph.1997.5776>. Citado 2 vezes nas páginas 69 e 70.

MÜLLER, M.; CHARYPAR, D.; GROSS, M. Particle-based fluid simulation for interactive applications. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003. (SCA '03), p. 154–159. ISBN 1-58113-659-5. Disponível em: <http://dl.acm.org/citation.cfm?id=846276.846298>. Citado 5 vezes nas páginas 9, 50, 51, 63 e 65.

NETO, A. P. *Uma abordagem lagrangeana para simulação de escoamentos de fluidos viscoplásticos e multifásicos*. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, 2007. Citado 11 vezes nas páginas 12, 50, 52, 53, 57, 58, 59, 60, 74, 79 e 83.

NEVES, R. C. *Os Quatérnios de Hamilton e o Espaço*. 112 p. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro – Instituto de Matemática, 2008. Citado 2 vezes nas páginas 44 e 45.

ONDERIK, J.; CHLÁDEK, M.; DURIKOVIC, R. Sph with small scale details and improved surface reconstruction. In: *Proceedings of the 27th Spring Conference on Computer Graphics*. New York, NY, USA: ACM, 2013. (SCCG '11), p. 29–36. ISBN 978-1-4503-1978-2. Citado 3 vezes nas páginas 9, 63 e 65.

PAIVA, A. et al. Simulação de fluidos sem malha: Uma introdução ao método sph. In: *27º Colóquio Brasileiro de Matemática*. [S.l.]: Instituto Nacional de Matemática Pura e Aplicada, 2009. Citado 4 vezes nas páginas 39, 41, 47 e 61.

PINTO, W. J. N. *Aplicação do método lagrangiano SPH (Smoothed Particle Hydrodynamics) para a solução do problema das cavidades*. 129 p. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2013. Citado 2 vezes nas páginas 21 e 36.

RAPP, B. Analytical solutions to poiseuille flow problems in different geometries. In: _____. *Microfluidics: Modelling, Mechanics and Mathematics*. [S.l.]: Elsevier, 2017. p. 323–349. ISBN 9781455731411. Citado na página 68.

RIBEIRO, M. d. L.; NUNES, F. L. S. Breast palpation simulation with haptic feedback: Prototype and initial results. In: *2014 XVI Symposium on Virtual and Augmented Reality*. [S.l.: s.n.], 2014. p. 268–276. Citado na página 22.

SANDIM, M. et al. Boundary detection in particle-based fluids. *Comput. Graph. Forum*, The Eurographs Association & John Wiley & Sons, Ltd., Chichester, UK, v. 35, n. 2, p. 215–224, maio 2016. ISSN 0167-7055. Disponível em: <https://doi.org/10.1111/cgf.12824>. Citado 3 vezes nas páginas 9, 64 e 65.

SCHECHTER, H.; BRIDSON, R. Ghost sph for animating water. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 31, n. 4, p. 61:1–61:8, jul. 2012. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/2185520.2185557>. Citado na página 48.

SHAHRIARI, S. et al. Smoothed particle hydrodynamics method applied to pulsatile flow inside a rigid two-dimensional model of left heart cavity. *International Journal for Numerical Methods in Biomedical Engineering*, v. 28, n. 11, p. 1121–1143, 2012. Citado 5 vezes nas páginas 29, 30, 31, 33 e 34.

SHAO, S.; LO, E. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources*, n. 26, p. 787–800, 2003. Disponível em: <http://hdl.handle.net/10454/3258>. Citado na página 75.

SHAO, X.; WANG, B.; ZHAO, M. Unified particle-based coupling algorithm for stable soft tissue-blood interaction. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. [S.l.: s.n.], 2016. p. 1–6. Citado 4 vezes nas páginas 31, 32, 33 e 34.

SHI, W.; ZHENG, M.; LIU, P. X. Virtual surgical bleeding simulation with navier-stokes equation and modified smooth particle hydrodynamics method. In: *2017 IEEE International Conference on Information and Automation (ICIA)*. [S.l.: s.n.], 2017. p. 276–281. Citado 3 vezes nas páginas 22, 31 e 52.

SOLENTHALER, B.; PAJAROLA, R. Predictive-corrective incompressible sph. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 28, n. 3, p. 40:1–40:6, jul. 2009. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/1531326.1531346>. Citado na página 63.

SOLENTHALER, B.; SCHLÄFLI, J.; PAJAROLA, R. A unified particle model for fluid–solid interactions: Research articles. *Comput. Animat. Virtual Worlds*, John Wiley and Sons Ltd., Chichester, UK, v. 18, n. 1, p. 69–82, fev. 2007. ISSN 1546-4261. Disponível em: <http://dx.doi.org/10.1002/cav.v18:1>. Citado na página 63.

SOUTO, H.; GóES, J.; GóES, M. Simulação numérica de escoamentos com superfícies livres empregando o método weakly compressible smoothed particle hydrodynamics (wcsph) e o processamento em paralelo com unidades de processamento gráfico e a api cuda. *Ciência e Natura*, v. 39, n. 3, p. 569–594, 2017. ISSN 2179-460X. Disponível em: <https://periodicos.ufsm.br/cienciaenatura/article/view/24887>. Citado na página 86.

STEWART, J. *Cálculo – Volume 2*. 5. ed. [S.l.]: Pioneira Thompson Learning, 2006. Citado na página 40.

TANAKA, N.; TAKANO, T. Microscopic-scale simulation of blood flow using sph method. *International Journal of Computational Methods*, v. 02, n. 04, p. 555–568, 2005. Disponível em: <https://www.worldscientific.com/doi/abs/10.1142/S021987620500065X>. Citado na página 22.

TORI, R. et al. Interlab: Interactive technologies laboratory. *SBC Journal on Interactive Systems*, v. 2, n. 2, p. 94–97, 2011. Citado na página 22.

VASCO, J. R. G. *Desenvolvimento de software utilizando a técnica sph (smoothed particle hydrodynamics) na geração de ondas de submersão*. 124 p. Tese (Doutorado) —

Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2014. Citado 7 vezes nas páginas 21, 37, 38, 39, 48, 54 e 56.

VUKCEVIC, V.; WERNER, A.; DEGIULI, N. Application of smoothed particle hydrodynamics method for simulating incompressible laminar flow. *Transactions of Fomena*, v. 36, p. 1–12, 01 2012. Citado na página 69.

XIAO, C. et al. Real-time and authentic blood simulation for surgical training. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2017. p. 6832–6837. Citado 5 vezes nas páginas 22, 30, 31, 32 e 33.

YE, T.; PHAN-THIEN, N.; LIM, C. Particle-based simulations of red blood cells - a review. *Journal of Biomechanics*, World Scientific Publishing Co., v. 49, n. 1-2, p. 2255–2266, 2016. Citado 4 vezes nas páginas 23, 26, 27 e 31.

YU, J.; TURK, G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 32, n. 1, p. 5:1–5:12, fev. 2013. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/2421636.2421641>. Citado 3 vezes nas páginas 9, 63 e 65.

ZHU, Y.; BRIDSON, R. Animating sand as a fluid. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 24, n. 3, p. 965–972, jul. 2005. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/1073204.1073298>. Citado 3 vezes nas páginas 9, 63 e 65.